Fire fighting Vehicle using Arduino, Auto fire chaser & Extinguisher

Dhushara Subodhini Wijemanna

ITBIN/2110/0120

Faculty of Information Technology

Horizon Campus

Supervisor

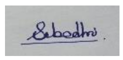
Ms. Samadhi Dewmini

June – 2025

This project proposal is submitted in partial fulfillment of the requirement of the degree of Bachelor of Science (Hons) in Information Technology of the Horizon Campus

Declaration

"I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Student: 

Date: …. / …. /.....

Name of Student: W.P.D.S.Wijemanna

Countersigned by: Signature of Supervisor(s): ………………………

Date: …. / …. /.....

Name(s) of Supervisor(s): Ms.Samadhi Dewmini

Abstract

Fire hazards are among the most destructive and unpredictable threats faced by society, often resulting in significant loss of life, property, and environmental damage. In many situations, especially in hazardous or inaccessible areas, manual firefighting may not be feasible or safe. To address this challenge, this project presents the development of an autonomous firefighting robot designed to detect and extinguish fire without human intervention. The proposed system, titled "Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher," aims to provide a cost-effective, efficient, and reliable solution to assist in early fire detection and suppression.

The robot is built on a mobile chassis controlled by an Arduino Uno microcontroller, which acts as the central processing unit of the system. A flame sensor is used to detect the presence and direction of fire. Once the sensor identifies a flame, the robot automatically navigates toward the source using simple decision-making logic and activates a water-pumping mechanism to extinguish the fire. The robot's motion is driven by DC motors, and the water is sprayed using a small submersible pump controlled by a relay module. The entire system is powered by a rechargeable battery pack, making it fully portable and suitable for both indoor and controlled outdoor environments.

The design and development process involved both hardware and software integration, including sensor calibration, motor driver control, and algorithm development to interpret sensor data and guide the robot's movement. The project emphasizes affordability and ease of assembly, using commonly available electronic components to ensure that the robot can be replicated or scaled up by others for educational, research, or practical applications.

Extensive testing was conducted in controlled environments to assess the robot's performance in fire detection accuracy, mobility, response time, and extinguishing effectiveness. Results indicated that the robot could reliably identify flame sources and successfully extinguish small-scale fires within a short response time. While the current prototype is limited to basic obstacle handling and operates best on flat surfaces, the results demonstrate strong potential for further enhancement.

This project contributes to the growing field of autonomous robotic systems for disaster management. By offering a simple yet effective prototype, it lays the groundwork for more advanced firefighting robots equipped with multiple sensors (e.g., gas, temperature, smoke), wireless communication, and improved navigation systems such as GPS or infrared cameras. With continued development, such systems could be deployed in real-world applications to improve safety and response times during fire emergencies, particularly in industrial plants, storage facilities, and residential areas.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to **Ms. Samadhi Dewmini** my project supervisor, for their invaluable guidance, encouragement, and constructive feedback throughout the development of this project. Their support has been instrumental in the successful completion of this thesis.

I would also like to thank the **Faculty of Information Technology at Horizon Campus, Malabe**, for providing the necessary facilities and academic environment that enabled me to conduct this research and practical implementation effectively.

My sincere appreciation goes to my colleagues and friends who contributed their insights, shared resources, and provided motivation during the challenging phases of this project.

A heartfelt thanks to my family for their constant support, patience, and understanding. Their encouragement has always been a source of strength for me.

Lastly, I extend my gratitude to all those who directly or indirectly supported me in the completion of this project titled **"Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher."** Their contributions, no matter how big or small, played a significant role in achieving the project's goals.

Contents

Abstract .... **iii**

Acknowledgement .... **iv**

Chapter 1 Introduction **1**

1.1 Background of the study 1

1.2 Problem Statement 2

1.3 Motivation and Significance of the project 3

1.4 Aim (s) and Objectives of the project 4

1.4.1 Aim 4

1.4.2 Objectives 4

1.5 Scope and Limitation of the project 5

1.5.1 Scope 5

1.5.2 Limitation 5

1.6 Chapter Outline 6

Chapter 2 Literature Review **7**

2.1 Introduction 7

2.2 Background on Firefighting Robots 7

2.3 Autonomous Firefighting Technologies 8

2.4 Role of Arduino in Robotics 8

2.5 Sensor Technologies for Fire Detection 9

2.6 The Navigation Mechanisms and Fire Chasing 10

2.7 Fire Extinguishing Methods 10

2.8 Issues and Limitations 11

2.9 Future Directions and Research Opportunities 12

Chapter 3 Analysis **13**

3.1 Introduction 13

3.2 Feasibility Study 13

3.2.1 Technical Feasibility 13

3.2.2 Financial Feasibility 14
3.2.3 Economic feasibility 14
3.2.4 Operational feasibility 14
3.3 Fact-Finding Technologies and User Stories 15
3.4 Requirements Analysis 15
3.5 Technical Requirements 15
3.6 Functional Requirements 16
3.7 Non-Functional Requirements 16
3.8 System Development Methodology 16
3.8.1 Input-Process-Output Framework 16
3.8.2 Software Development Life Cycle 17
3.8.3 Development Methodology 17
Chapter 4 Design 20
4.1 Introduction 20
4.2 System Design Process 21
4.2.1 Use case Diagram 22
4.2.2 Class Diagram 23
4.2.3 Block Diagram 24
4.2.4 Activity Diagram 25
4.2.5 Circuit Diagram 26
4.3 Interface Design 27
4.3.1. Design principles 27
4.4. Hardware Components 29
Chapter 5 System Development 36
5.1 System Overview 36
5.2 Development Environment 37
5.3 Tools and Technologies 38
5.4 Development Architecture 39
5.4.1 Overview of the System Architecture 39

5.4.2 Architecture Diagram 40
5.5 Major Code Segment 41
Chapter 6 Testing and Evaluation 46
6.1 Introduction 46
6.2 Testing Procedure 46
6.2.1. Unit Testing 47
6.2.2 Integrating Testing 47
6.2.3. Real-World Scenario Testing 48
6.3 Test Plan and Test Case 48
6.3.1 Test Plan 48
6.3.2 Test Case 49
6.4 Test Data and Test Result 51
6.5 Acceptance Testing 51
Chapter 7 Conclusion and Future Work 52
7.1 Conclusion 52
7.2 Future Work 53
Chapter 8 References .... Error! Bookmark not defined.
References 54
Appendix A – System Documentation 57
Appendix B – Design Documentation 58
Appendix E – Test Results 60
Appendix F - Code Listing 60
Glossary 66
Index .... Error! Bookmark not defined.

List of Figures

FIGURE 2.1 HARDWARE MODEL OF PROPOSED SYSTEM 7
FIGURE 2:2 ARCHITECTURE MODEL OF THE PROPOSED SYSTEM 8
FIGURE 2.3 FLOW CHART OF PROPOSED SYSTEM 9
FIGURE 2.4 FIRE PROGRESSION 9
FIGURE 2.6 CIRCUIT DIAGRAM PROPOSED SYSTEM 10
FIGURE 2.5 PROPOSED SYSTEM 11
FIGURE 2.7 EXTINGUISHER METHODS 11
FIGURE 2.8 ARCHITECTURE OF ARDUINO UNO BOARD 12
FIGURE 3.1 ITERATIVE MODEL 17
FIGURE 4.1 USE CASE DIAGRAM 23
FIGURE 4.2 CLASS DIAGRAM 24
FIGURE 4.3 BLOCK DIAGRAM 24
FIGURE 4.4 ACTIVITY DIAGRAM 25
FIGURE 4.5 CIRCUIT DIAGRAM 27
FIGURE 4.6 ARDUINO UNO BOARD 29
FIGURE 4.7 FLAME SENSOR 30
FIGURE 4.8 BO MOTORS AND WHEELS 30
FIGURE 4.9 MOTOR DRIVER 31
FIGURE 4.10 BREAD BOARD 31
FIGURE 4.11 SERVO MOTOR 32
FIGURE 4.12 WATER PUMP WITH PIPE 32
FIGURE 4.13 18650 ×2 3.7V LITHIUM-ION BATTERIES 33
FIGURE 4.14 JUMPER WIRES 33
FIGURE 4.15 TRANSISTOR 34
FIGURE 4.16 CAPACITOR 34
FIGURE 4.17 RESISTOR 34
FIGURE 4.18 RELAY MODULE 35
FIGURE 5:1 NAVIGATION/ SYSTEM STRUCTURE / SYSTEM OVERVIEW 36
FIGURE 5.2 ARCHITECTURAL DIAGRAM 40
FIGURE 5.3 MAJOR CODE I 41
FIGURE 5.4 MAJOR CODE II 42
FIGURE 5.5 MAJOR CODE III 42
FIGURE 5.6 MAJOR CODE IV 43
FIGURE 5.7 MAJOR CODE V 43
FIGURE 5.8 MAJOR CODE VI 44
FIGURE 5.9 CODE COMPILING 44
FIGURE 5.10 DONE UPLOADING FOR ARDUINO UNO BOARD 45
FIGURE 0.1 USE CASE DIAGRAM FOR FIRE FIGHTING ROBOT 58
FIGURE 0.2 CLASS DIAGRAM FOR FIRE FIGHTING ROBOT 58
FIGURE 0.3 BLOCK DIAGRAM FOR FIRE FIGHTING ROBOT 59
FIGURE 0.4 ACTIVITY DIAGRAM FOR FIRE FIGHTING ROBOT 59

List Of Table

TABLE 3.1 FINANCIAL FEASIBILITY 14
TABLE 5-1 DEVELOPMENT ENVIRONMENT 37
TABLE 5-2 TOOLS AND TECHNOLOGIES 38
TABLE 6.1 TEST CASE 1 - FLAME DETECTION MODULE 49
TABLE 6.2 TEST CASE 2 - NAVIGATION TOWARDS FIRE 49
TABLE 6.3 TEST CASE 3 - FIRE EXTINGUISHING MODULE 50
TABLE 6.4 TEST CASE 4 - ARDUINO CONTROL AND COMMUNICATION 50
TABLE 6.5 TEST DATA AND TEST RESULT 51
TABLE 0.1 TEST RESULT 60

Chapter 1 Introduction

1.1 Background of the study

Fire is one of the most destructive forces that can lead to devastating consequences for individuals, communities, and the environment. Every year, countless lives are lost, and significant property and environmental damage occurs due to uncontrolled fires. Fire incidents often escalate quickly, making rapid response a critical factor in minimizing loss and damage. However, most of the traditional firefighting methods rely on human intervention, which involves severe risks to firefighters operating in dangerous environments such as high-temperature zones, smoke-filled areas, and unstable structures.

In this regard, integration of robotics and automation in firefighting operations has emerged as a promising solution for such challenges. The concept of firefighting robots is rooted in the need to improve safety and efficiency in fire emergency responses. With the deployment of robots that can detect and extinguish fires, the risks to human firefighters will be minimal, with the added advantage of speed and precision in putting out a fire. This will not only be safer for the operation but also reduce the time needed to locate and contain fires in inaccessible or hazardous locations.

Firefighting robotics has drawn a lot of interest in the last few years because of the advances in sensors, microcontrollers, and autonomous navigation systems. Arduino, as an open-source microcontroller platform, has been widely used for such applications due to its versatility, affordability, and ease of programming. Integrating flame sensors with Arduino-based robots along with the mobility mechanism will enable them to locate the fire, move towards it, and extinguish it effectively.

This project, "Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher," aims at developing an autonomous robot designed for handling small-scale fire incidents. The system detects the source of a fire with the help of a flame sensor and processes the input through an Arduino microcontroller for navigation control. The robot will be able to extinguish the fire once it reaches the location.

The study will also attempt to show the practical feasibility of robotics in fire safety from the point of view of accessibility, simplicity, and effectiveness. This innovation responds to the increasing demand for firefighting solutions that are much safer and contributes to the broader objective of enhancing automation in disaster management. This project emphasizes the role of technology in creating a safer environment while reducing human intervention during emergencies.

1.2 Problem Statement

Fire hazards remain a significant threat to human life, property, and the environment. Fires can occur unexpectedly in residential, industrial, and commercial settings, often spreading rapidly and causing irreversible damage. The ability to respond promptly to fire outbreaks is critical in minimizing their impact, but traditional firefighting approaches largely depend on human firefighters who face substantial risks when entering dangerous environments. These include risks from severe heat, smoke inhalation, toxic gases, and physical injuries due to weak structures. Even with increased development in firefighting appliances and personal protective equipment, the health of firefighters is still the biggest apprehension.

Further, fires that happen in inaccessible or dangerous areas, like confined spaces, chemical plants, and disaster-prone areas, require more challenges. There, the delay in reaching the source of fire creates a bigger mess with more potential loss of life. These constraints in human response times, coupled with the hazardous nature of the operation, emphasize the need for some kind of innovative intervention through automation in firefighting.

The second critical issue is the existing costs and complexities of fire detection and suppression systems. With highly advanced systems available in the market, most require very expensive infrastructure modification. The systems are just out of reach for small-scale applications or low-resource, small-scale settings. Especially in developing countries, with very limited resources and access to technology.

In this context, the integration of robotics and automation offers a viable solution to improve fire safety. Firefighting robots equipped with sensors and extinguishing mechanisms can perform autonomously in hazardous environments, thus minimizing the risks to human firefighters. However, most of the current robotic systems are either too complex, expensive, or designed for industrial applications, which limits their broader diffusion.

This project aims to resolve these challenges by presenting the design of an Arduino-based firefighting robot for fire detection and extinguishing autonomously. The flame sensor-dependent robot detects fire and navigates towards it for a quick response. The design is cost-effective and has simple implementation, hence finding its perfect use in small-scale applications and areas where advanced systems may not be available.

The problem to be solved is the need for a low-cost, efficient, and reliable system to identify and extinguish fire incidents with minimal human involvement under hazardous conditions. Emphasis on accessibility and ease of use will be the guiding concerns in this project in terms of providing a practical means that can be applied residentially, commercially, or industrially to improve the state of fire safety while eliminating the need for human firefighters as much as possible in risky circumstances.

1.3 Motivation and Significance of the project

The motivation for this project arises from the growing incidents of fire outbreaks and their awful effects. Fires suddenly break out in homes, workplaces, or public places; these are usually very fast-spreading and may have disastrous results on life and property. Traditional firefighting involves a lot of human intervention, which exposes the firefighters to severe risks like extreme temperatures, toxic smoke, and physical injuries. These challenges show a need for innovative approaches in fire response while ensuring safety to the responders.

The rise of automation and robotics provides an excellent opportunity that can help in mitigating many of these issues amply. By taking advantage of modern technologies, autonomous systems can be devised that might handle hazardous jobs and would reduce the role of man in dangerous scenarios. Undeniably the most ambitious idea is creating a firefighting robot independently detecting fire and extinguishing it-marrying three concepts at once: safety, efficiency, and the use of advanced technology.

Also, the affordability and versatility of platforms such as Arduino have opened new avenues for the development of practical and cost-effective robotic solutions. This project is motivated by the need to make fire safety technology more accessible, especially for small-scale applications or in resource-poor regions. The ability to save lives and reduce fire-related damages with an autonomous system is one of the driving forces behind this initiative.

This project bears great relevance in solving critical problems that have to do with fire safety. The construction of the firefighting autonomous robot is very relevant and contributes much toward the standards of safety at places that are highly risky or could be difficult for human entry. This could be particularly important, ensuring timely detection and extinction, which prevents the spread of fire:

The use of an Arduino-based platform enhances the practicality of the project through an inexpensive and easily replicable solution. This makes the system accessible not only to industries but also to households, small businesses, and community centers. Its simplicity and adaptability are particularly important in low-resource settings where advanced fire safety systems are often unavailable.

The project also highlights the role of technology in disaster management and safety, showing how robotics can be used to solve real-life problems. It will also encourage further research and development in the field of autonomous systems, thus encouraging innovation in creating safer and more efficient solutions for different applications. Ultimately, this project aspires to contribute to a future wherein technology plays a central role in saving lives and properties from fire hazards.

1.4 Aim (s) and Objectives of the project

1.4.1 Aim

This project aims to make an autonomous firefighting robot that is based on Arduino, which can find and stop the fire very well to avoid human involvement in dangerous fire-fighting operations. The system is supposed to present a solution for the quick and effective extinguishing of small fire tragedies, the increase of security as well as the decreasing the response time, residential, commercial, and industrial buildings in cases of risk fire."

1.4.2 Objectives

- Develop a Fire Detection Mechanism

Integrate a flame sensor to accurately detect fire sources and trigger the robot's response system.

- Implement Autonomous Navigation

Design a navigation system that allows the robot to move towards the fire source using real-time sensor feedback and avoid obstacles.

- Incorporate an Extinguishing Mechanism

Equip the robot with a fire suppression system capable of extinguishing small-scale fires effectively upon reaching the fire source.

- Utilize Arduino for Control and Processing

Employ the Arduino microcontroller as the core processing unit to handle inputs from sensors and control the robot's movements and actions.

- Ensure Cost-Effectiveness and Simplicity

Focus on designing a system that is affordable and simple to implement, making it accessible to a wide range of users, including those in low-resource settings.

- Enhance Safety in Firefighting Operations

Minimize the need for human intervention in hazardous fire scenarios by automating fire detection and suppression tasks.

- Evaluate Performance in Real-World Scenarios

Test the robot in controlled environments to assess its accuracy in fire detection, navigation efficiency, and extinguishing effectiveness.

- Promote Innovation in Fire Safety Technology

Demonstrate the potential of robotics and automation in addressing critical challenges in fire safety and inspire further advancements in the field.

The project aims to produce a functional and steadfast robot firefighter that will contribute decisively to fire safety and disaster control in potential risky urban environments by using traditional methods and unsafe activities.

1.5 Scope and Limitation of the project

1.5.1 Scope

The scope of this work deals with the design and deployment of an autonomous firefighting robot based on Arduino. It should be able to operate within small-scale fires that have just started, hence is meant for a house, small office, or laboratory. The system has been designed with a flame sensor for fire detection, an Arduino microcontroller to process and control the functions, and a mobility system that navigates the system towards the fire source. When the fire is detected, the robot extinguishes it by using a compact extinguishing mechanism.

The project highlights the possible incorporation of robotics and automation for fire safety applications: access, affordability, simplicity. As stated, it is to operate independently where human contact has to be lessened, being ideal to work where a normal and traditional firefighting is unrealistic or just dangerous. Apart from these factors, it is developed so that new enhancements, for the purpose of bigger and more complex firefighting jobs, can further be implemented and modified over time.

1.5.2 Limitation

- Fire Detection Range

The robot uses a flame sensor only, with a limited detection range that is good mainly for the detection of open flames. This will probably result in poor performance in the detection of smoke or heating that indicates the presence of fire in its early stages.

- Scale of Fires

The robot is designed for small-scale fires and may not be effective in handling large-scale or rapidly spreading fire incidents. Its extinguishing mechanism is optimized for minor fire outbreaks only.

- Navigation Constraints

The navigation system of the robot is rudimentary and may face challenges in highly cluttered or complex environments. It is also not equipped with advanced mapping or localization features like GPS or LiDAR.

- Environmental Conditions

The robot is intended for controlled indoor environments and may not function effectively in outdoor settings with high wind, rain, or extreme temperature variations.

- Limited Battery Life

The robot's operational duration is constrained by the battery life of its components, which may require frequent recharging for extended usage.

- No Multi-Sensory Input

The system uses only one flame sensor and does not employ other sensors, such as smoke detectors, temperature sensors, or cameras, which could add to its capability in the detection and analysis of fire situations.

- Manual Refilling of Extinguishing Mechanism

The extinguishing mechanism needs to be refilled or recharged manually after every operation, thereby limiting the robot's capabilities in attending to successive fire incidents.

Despite these limitations, the project provides the base framework for developing firefighting robots that are cost-effective and accessible, with potential upgrading and enhancement in the future that will increase its capabilities beyond the constraints it has now.

1.6 Chapter Outline

This chapter begins with a discussion regarding fire hazards and limitations of traditional firefighting, with emphasis on threats to firefighters. It introduces robotics and automation as a safer, efficient, and effective alternative, with emphasis on Arduino-based firefighting robots. Problem statement describes shortcomings of current fire suppression methods, with a concentration on a low-cost, automated approach. Motivation highlights growing numbers of fires and automation's role toward enhancing fire response.

The significance of the project is explained, outlining how it can contribute to fire safety. The aim and objectives of the project are then explained, with an emphasis on fire detection, navigation, and fire suppression. Finally, project scope and limitations are covered, with emphasis on its usefulness for small-scale indoor fires and potential areas for further development.

Chapter 2 Literature Review

This chapter offers an overview of literature studies on the programs and effectiveness of numerous types of fire fighting vehicle using Arduino, auto fire chaser and Extinguisher.

2.1 Introduction

Fire-related risks are one of the greatest threats to human life and property around the world. Fire-related incidences extinguish large numbers of resources associated with it and, more importantly, claim many important human lives; this is according to the NFPA. These risks elicited from fire-related accidents form the basis for the development of high technological interventions that target the mitigation of these risks. One such solution would be the invention of autonomous, segregating robotic firefighters that can quickly and efficiently respond to detect and extinguish a fire, reducing human susceptibility in dangerous situations [1].

Robotics development, coupled with advancing sciences and sensor technologies used in microcontroller systems, together may have the eventual consequence of finally making it possible to develop a sophisticated and reliable chain of firefighting robots [2]. The robots are meant to navigate independently through fire-prone environments, locate flames, and take action towards extinguishing the fire. Some research has focused on using microcontrollers, including Arduino, for the development of inexpensive and programmable firefighting robots [3].

2.2 Background on Firefighting Robots

Robotics development in firefighting over the last few decades has improved due to breakthroughs in new technologies. The first firefighting robots were conceptualized to be used remotely, with humans at a safe distance guiding the movement of the robot [4]. Therefore, these robots initially focused on user awareness in hazardous environments and comprised mainly cameras and basic fire detection sensors. As technology advanced, autonomous robots capable of moving around fire scenes and identifying fire sources were developed [5].

The firefighting robot provides several benefits over traditional applications. It can operate in environments that would be dangerous for human firefighters due to extreme heat, toxic smoke, or structural collapse. This significantly reduces the risk to human life [6]. Additionally, robots can enter spaces that are inaccessible to humans, ensuring effective fire extinguishment [7].

Figure 2.1 Hardware Model Of proposed system

2.3 Autonomous Firefighting Technologies

Early research on autonomous firefighting technologies focused on improved navigation and fire detection capabilities. Infrared sensors and temperature sensors were introduced in studies during the late 20th century, enabling the identification of fire sources [8]. Advances in obstacle detection sensors and navigation systems later enhanced the capabilities of autonomous firefighting robots. The advent of microcontrollers, such as Arduino, further accelerated the development of cost-effective and flexible firefighting robots [9].

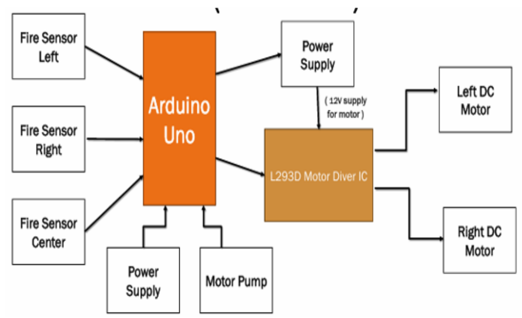


Figure 2:2 Architecture Model of the proposed system

2.4 Role of Arduino in Robotics

Arduino is an open-source electronics platform that facilitates robotic development due to its ease of use in both hardware and software. It utilizes a microcontroller programmed to perform specific tasks, making it a flexible tool for developing a wide range of robotic applications [10]. Due to its simplicity, low cost, and large community support, Arduino has become a popular platform for many researchers and engineers interested in building autonomous systems, including firefighting robots [11].

The flexibility of Arduino allows developers to integrate different kinds of sensors, actuators, and communication modules as needed. This adaptability makes it particularly useful for designing firefighting robots that require real-time fire detection, navigation, and extinguishing capabilities [6].

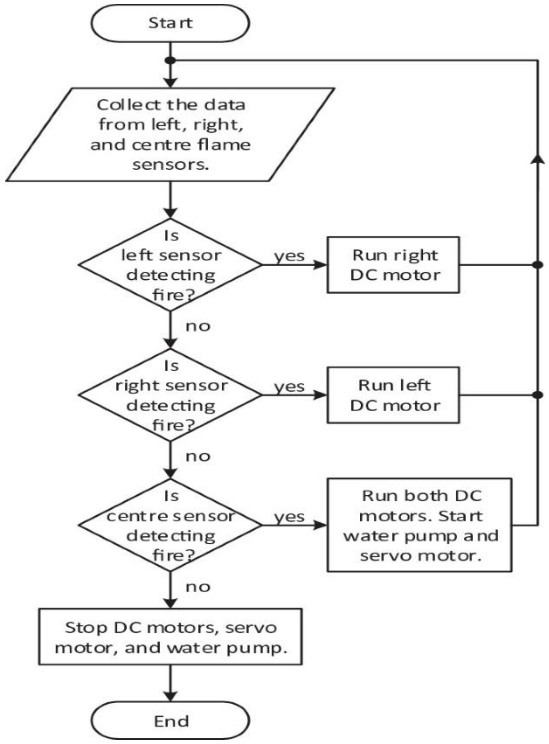


Figure 2.3 Flow Chart of proposed system

2.5 Sensor Technologies for Fire Detection

Firefighting robots use various sensors for fire detection. Common sensors include infrared (IR) sensors, temperature sensors, gas sensors, and flame sensors. Each sensor type provides unique capabilities, such as detecting heat radiation, measuring ambient temperature changes, identifying combustion gases, and recognizing specific wavelengths of flame light [8].

Studies have focused on integrating multiple sensors with Arduino-based platforms to enhance fire detection accuracy. For example, research by Kaur and Kaur (2018) demonstrated that combining IR and temperature sensors improves the reliability of fire detection and reduces false alarms [6].

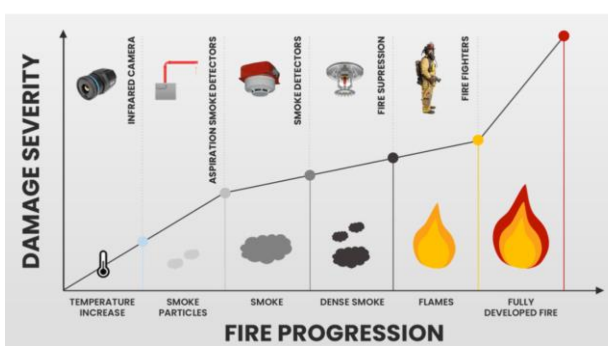


Figure 2.4 Fire Progression

2.6 The Navigation Mechanisms and Fire Chasing

Firefighting robots use various autonomous navigation methods to efficiently reach the fire source. Common techniques include line following, obstacle detection, and GPS-based navigation [5].

- Line Following – Infrared sensors detect pre-marked paths, making it an effective and low-cost method for indoor navigation [4].
- Obstacle Avoidance – Robots use ultrasonic sensors, infrared sensors, and LIDAR to detect and maneuver around obstacles [7].
- GPS-Based Navigation – In large indoor or outdoor environments, GPS data improves the accuracy of robot localization [9].

Studies have investigated the effectiveness of different navigation algorithms. Research by Tanaka and Komatsu (2012) highlighted the advantages of combining line-following and obstacle-avoidance strategies for improved navigation [9]. Similarly, Hwang and Lee (2015) demonstrated that fuzzy logic and neural networks enhance robot navigation in complex environments [10].

Figure 2.6 Circuit Diagram Proposed System

2.7 Fire Extinguishing Methods

Firefighting robots employ various extinguishing techniques, including water spray, CO_2, dry powder, and foam-based suppression systems [2].

- Water Spray – Commonly used for Class A fires involving wood, paper, and fabric [3].
- CO_2 Extinguishers – Effective for Class B (flammable liquids) and Class C (electrical) fires [4].
- Dry Powder – Useful for various fire types, including electrical fires [7].
- Foam Extinguishing Systems – Prevent flames from receiving oxygen, suitable for liquid fires [8].

Studies have explored the integration of extinguishing systems with Arduino-powered robots. Elijah and Adebayo (2018) demonstrated that CO_2 suppression systems controlled by Arduino improve efficiency and accuracy in fire suppression [7].

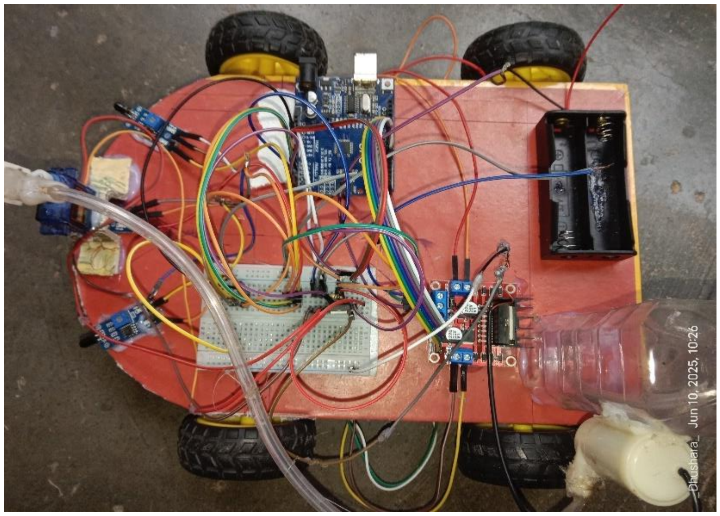


Figure 2.5 Proposed System

2.8 Issues and Limitations

Firefighting robots face several technical challenges, including sensor accuracy, navigation precision, and power consumption [9]. Fire environments are particularly harsh, with smoke affecting sensor reliability and extreme heat damaging electronic components [11].

Researchers have identified the need for improvements in sensor fusion, heat-resistant materials, and power efficiency. Enhancing autonomous learning capabilities would allow robots to dynamically adapt to real-world fire scenarios [10].

Figure 2.7 Extinguisher Methods

2.9 Future Directions and Research Opportunities

The future of firefighting robots is expected to be shaped by advancements in artificial intelligence, machine learning, and improved sensor technologies [8].

Emerging trends include:

- Multi-Spectral Sensors and Thermal Imaging – Enabling more precise and rapid-fire detection [7].
- Enhanced Navigation Systems – Using LIDAR and simultaneous localization and mapping (SLAM) for improved movement in unstructured environments [6].
- Collaborative Multi-Robot Systems – Allowing teams of robots to handle large-scale fires [11].

Conclusion

This literature review highlights the advancements and challenges associated with Arduino-based firefighting robots. The integration of sensor technologies, navigation systems, and extinguishing mechanisms has made autonomous firefighting solutions increasingly effective. Future research should focus on improving sensor fusion, AI-based decision-making, and the overall resilience of firefighting robots [8].

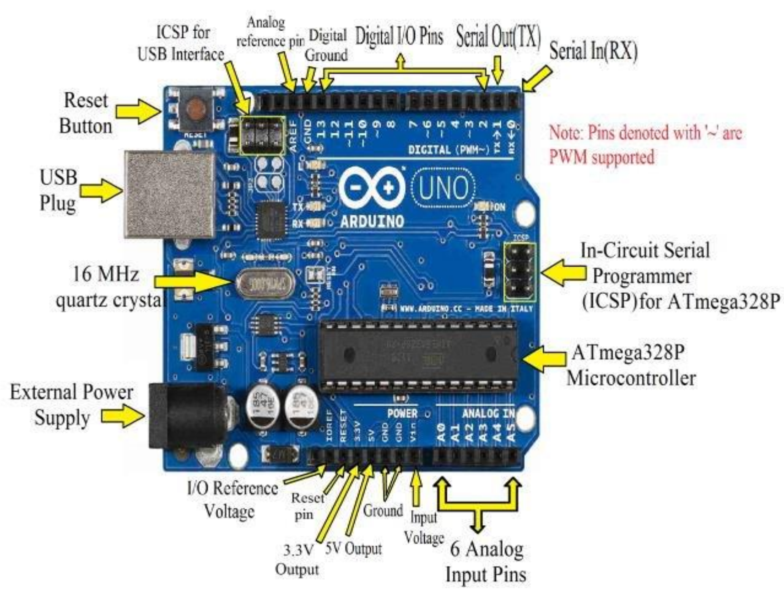


Figure 2.8 Architecture of Arduino Uno Board

Chapter 3 Analysis

3.1 Introduction

The analysis phase of this thesis involves core functionalities, design considerations, and performance evaluation of the Arduino-based firefighting robot. This chapter therefore presents an in-depth review of all components, technologies, and methods adopted for autonomous fire detection and extinguishing. We analyze how sensor technologies for fire detection, navigation systems, and extinguishing mechanisms have been integrated into ensuring that the efficiency, reliability, and limitations of the robot under various conditions, both environmental and operational, are explored.

In order to ensure that the approach is structured, the analysis focuses on three major aspects: fire detection accuracy, efficiency in navigation and obstacle avoidance, and effectiveness of extinguishing methods. The performance of various sensor types, such as flame, infrared, and temperature sensors, is compared to understand their accuracy and responsiveness in detecting fire under real-world scenarios. The ability of the navigation system to adapt to dynamic environments in order to avoid obstacles and reach the sources of fire effectively is critically assessed. Besides, operational efficiency regarding the extinguishing systems, such as water sprays, CO_2, and other suppression methods, for various classes of fires is considered.

The chapter also explores the shortcomings that were observed during the implementation and testing phases, including sensor inaccuracies, environmental constraints, and challenges in power consumption. From these, the basis for the identification of areas of improvement and the proposal of recommendations for future enhancements are obtained. This analysis forms the cornerstone for validating the research objectives and contributes to the development of more efficient and reliable firefighting robots.

3.2 Feasibility Study

The feasibility study is intended to evaluate practically and determine the viability of the Arduino-based firefighting robot proposed, considering technical feasibility, economic feasibility, and operational viability.

3.2.1 Technical Feasibility

From the technical point of view, the feasibility of the system is considered in terms of the integration of Arduino with sensors, actuators, and extinguishing systems. The selection of components is justified by their compatibility, availability, and ability to meet the performance requirements for fire detection, navigation, and suppression. Adaptability of the robot's design to different fire scenarios and environments is also considered to ensure robustness and scalability.

3.2.2 Financial Feasibility

In a financial feasibility study, how much capital is needed to start the project, as well as the sources of funds, the rate of return on investment, and other financial factors.

Table 3.1 Financial Feasibility

Equipment	Number of pieces Amount	Amount (Rs)
Arduino Uno	1	Rs. 2100.00
Flame Sensors	3	Rs.2370.00
Bread Board	1	Rs. 300.00
Motor Driver	1	Rs. 890.00
Servo Motor	1	Rs. 500.00
Bo Motors And Wheel	4	Rs.1290.00
1860 6800mAh Battery	2	Rs.1760.00
Battery case	1	Rs. 300.00
Jumper Wire		Rs. 500.00
Switch	1	Rs. 140.00
Water Pump	1	Rs. 300.00
5V Single Relay Module	1	Rs. 300.00
Resistor/Diode	1/1	Rs. 150.00
Capacitor	3	Rs. 300.00
Transistor	1	Rs. 100.00
Total		**Rs. 11300.00**

3.2.3 Economic feasibility

Economic feasibility is analyzed with an estimation of the component cost, cost of assembling, and maintenance. Cost-effectiveness compared to manual conventional firefighting methods and other robotic solutions is done to expose the potential for wider application in resource-constrained environments. The major contribution of Arduino and all other components used in ensuring the project remains within constrained budgets without compromising functionality is their affordability.

3.2.4 Operational feasibility

Operational feasibility involves assessing the ease of deployment, the user-friendliness of the system, and operational reliability in real-world use. The ability of the robot to operate autonomously without much human intervention, resisting environmental factors such as high temperatures and smoke, should be energy-efficient. Besides the difficulties that may arise in practical implementation, the need to train operators, regular maintenance, and integration with protocols already in place for firefighting have been discussed.

In all, the feasibility study confirms the project's potential for successful implementation and identifies further optimizations that could enhance the effectiveness and sustainability of the intervention.

3.3 Fact-Finding Technologies and User Stories

Fact-finding in the Arduino-based firefighting robot required research into available technologies and the gathering of information from potential users to ensure that the functionalities of the system meet real-world requirements. This process looked at recent developments in sensor integration, navigation algorithms, and fire suppression mechanisms to identify technologies that would be most effective for the robot.

Sensor technologies were assessed in terms of their accuracy and reliability in fire detection under different scenarios. Infrared sensors, flame sensors, and temperature sensors were important components owing to their complementary capabilities for fire detection in different respects. Moreover, navigation technologies such as ultrasonic sensors, LIDAR, and GPS systems have been explored in the literature for obstacle avoidance and precise localization in dynamic environments. Extinguishing methods, including water sprays, CO_2 canisters, and dry powder mechanisms, were put up against compatibility with Arduino's control systems and effectiveness on many fire classes.

User stories were developed to represent all needs and expectations by users: firefighters, safety officers, and facility managers. These stories described where the robot could be deployed, such as confined spaces in industrial settings, extinguishing fires in hazardous areas, and giving real-time updates to command centers. These stories provided insight into guiding the design and functionality of the robot to meet practical requirements of ease of use, reliability, and adaptability for various fire scenarios.

The facts identified through this fact-finding process shaped the selection of technologies and the development of user-centric features, thus creating a robust and efficient firefighting solution.

3.4 Requirements Analysis

The requirement analysis outlines the major technical and functional requirements for the Arduino-based firefighting robot. In this stage, the major components, their interaction with each other, and system-level requirements to be established in order to achieve the objectives of the project will be determined.

3.5 Technical Requirements

Sensors: The robot should incorporate flame, infrared, and temperature sensors that are capable of providing actual fire detection and identification for different types of fires.

Navigation: Ultrasonic sensors, LIDAR, or GPS systems for obstacle detection and efficient path planning in dynamic environments.

Extinguishing Mechanisms: The system shall contain actuators for water sprays, CO_2 canisters, or dry powder dispensing to address various fire scenarios.

Processing Unit: The core processing unit shall be an Arduino microcontroller to manage data from sensors, control actuators, and implement navigation algorithms.

3.6 Functional Requirements

- The robot should be able to move itself toward the source of the fire while avoiding obstacles.
- It should detect and classify the type of fire to deploy appropriate extinguishing methods.
- Communication modules such as Bluetooth or Wi-Fi should be used to enable real-time monitoring and feedback.

The system should work reliably under extreme conditions, including high temperatures and low visibility.

3.7 Non-Functional Requirements

- Scalability: The design should support additional sensors or components for future upgrades.
- Durability: The components should be able to bear the extreme environmental conditions of heat, smoke, and debris.
- Energy Efficiency: The robot should be designed in such a way that it optimizes power consumption for extended operational time.
- Cost-Effectiveness: The system shall be developed within budgetary limits without compromising performance and reliability.

This requirement analysis gives a clear framework for design, development, and testing of the firefighting robot in such a way that all critical aspects are considered to meet user expectations and operational demands.

3.8 System Development Methodology

The development of the Arduino-based firefighting robot follows a structured approach informed by established software development and engineering methodologies. This section gives the steps taken to ensure successful implementation of the system.

3.8.1 Input-Process-Output Framework

The IPO model is adopted in the system design as follows:

- Input: Data from sensors that include flame sensors for the detection of fire, heat sensors for changes in temperature, and navigation sensors for the avoidance of obstacles.
- Process: The Arduino microcontroller will process the input data in determining fire presence, plotting navigation paths, and actuating extinguishing mechanisms.
- Output: The output would be the robot executing tasks like navigating towards the source of the fire, evading obstacles, extinguishing the fire, and sending alerts/logs.

3.8.2 Software Development Life Cycle

The development is to follow the SDLC model as it ensures systematic and organized development in:

- Requirement Gathering and Analysis: Functional and non-functional requirements were determined to guide development.
- System Design: The architecture, both hardware and software interfaces, is designed based on the requirement.
- Implementation: Program the Arduino microcontroller for sensor integration, navigation, and fire suppression functionality.
- Testing: Unit testing, integration testing, and system testing are done to validate system performance for different scenarios.
- Maintenance: Plans for post-deployment updates and calibration to ensure long-term reliability.

3.8.3 Development Methodology

This project utilizes the Agile methodology for its iterative development and improvement. The various components comprising sensors, navigation algorithms, and fire suppression mechanisms will be developed and tested in iterations. Agile will help accommodate feedback and flexibility due to changing requirements at any moment during the development of this project.

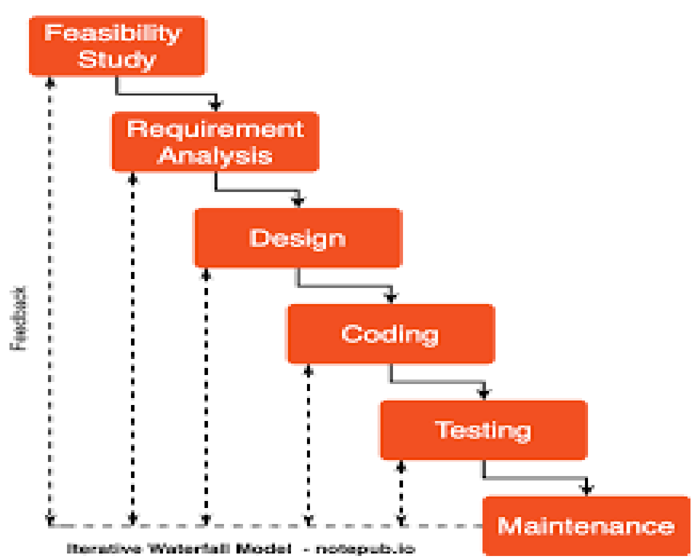


Figure 3.1 Iterative Model

3.8.3.1 Advantages of Iterative model

The Iterative Model is very suitable for developing a Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher as it allows continuous enhancement through repeated design, test, and refine phases. The key advantages of using the Iterative Model for this project are:

- Early Detection of Issues

The iterative approach enables frequent testing of the firefighting robot at each stage of development.

Defects in fire detection, navigation, and extinguishing systems can be found and addressed early in the process.

- Incremental Development for Managing Complexity

The functions of the robot, such as fire detection, navigation, and extinguishing, can be established and enhanced step by step.

Individual subsystems like sensor integration, motor control, and communication modules can be developed independently before being integrated.

- Adaptability and Flexibility

Changes in hardware (new Arduino boards, sensors, or actuators) or software (enhancements to fire detection algorithms) can easily be integrated in future iterations.

If a more effective fire suppression method (e.g., water vs. CO_2) is required, modifications can be made without restarting the project from scratch.

- Continual Performance Improvement

The robot's navigation and obstacle avoidance can be enhanced in multiple iterations using path optimization techniques (e.g., line-following and obstacle detection).

Iterative improvement minimizes false alarms in fire detection by improving sensor data fusion (IR, temperature, and flame sensors).

- Risk Reduction

Since the firefighting robot is deployed in risky scenarios, iterative testing ensures that it performs effectively in different fire scenarios prior to full deployment.

Potential failures (e.g., sensor inaccuracy, poor battery performance, or ineffective fire extinguishing) can be determined and addressed step by step.

- User Feedback Integration

If the robot is being designed for real-world firefighting support, input from fire safety experts can be incorporated at various points.

The accuracy, response time, and durability of the robot can be improved based on tests conducted in real-life scenarios.

- Cost-Effectiveness

With step-by-step implementation, the development team can commit resources incrementally rather than all at once.

Blocking major failures early prevents costly rework, which reduces the overall development cost.

- Real-Time Data and AI Enhancements

Later iterations can incorporate machine learning algorithms so that fire detection accuracy improves over time.

Data from previous test runs can be utilized to train the robot to differentiate between actual fires and false alarms (e.g., sources of heat or sunlight).

Chapter 4 Design

4.1 Introduction

This is the design phase of the project, covering the systematic development of the architecture and components of the Arduino-based firefighting robot. Previous requirements and analysis chapters will be translated into an extensive framework that will lead to implementation. By focusing on hardware and software integration in this chapter, it ensures that the functionality of the robot serves the purpose of its intended use in realistic environmental conditions for autonomous fire detection and suppression.

Central to the design is the system architecture that integrates several sensors, navigation algorithms, and fire suppression mechanisms. The robot design is actually built around the integration of these components in a manner that achieves seamless coordination to help meet the functional and non-functional requirements highlighted earlier. The placement and selection of sensors, such as flame, infrared, and temperature detectors, for instance, is considered crucial for perfect fire detection and response time. Schedules regarding navigation systems-obstacle avoidance and pathfinding algorithms-are elaborated carefully toward better mobility in dynamic conditions of medium features. The fire suppression system, be it with the use of water spray or CO2 canisters among others, fits into the design to work effectively and flexibly around different fire situations.

The physical design considerations go a long way in deciding the overall performance of the robot. The chassis layout, the placement of sensors and actuators, and selection of appropriate materials optimize regarding durability, stability, and efficiency of operations. More importantly, special attention is paid to weight distribution and compactness for better negotiation of the robot through confined spaces and challenging terrains.

Other interesting features of the design involve the user interface, from which the robot can be monitored and controlled remotely. The interface is designed to be highly intuitive, having real-time feedback like sensor data, system status, and alerts. This feature ensures ease in overseeing the activities of the robot by operators and intervening if need be.

The design process also puts in consideration scalability and maintainability. The architecture is structured to support future upgrades, such as the addition of more sensors or better algorithms, to keep it relevant to newer technologies.

The design phase fills the gap between theoretical analysis and practical implementation; thus, it lays the concrete foundation for the development of a robust, efficient, and autonomous firefighting robot with the capability for a wide range of fire safety challenges.

4.2 System Design Process

The system design process is a structured approach to developing an autonomous robotic system that has the capability to detect and extinguish fire efficiently. The process makes the system functionally correct as well as scalable, maintainable, and reliable.

The system design process is presented in the following phases,

1. Requirements Gathering: This stage is concerned with identifying the key requirements of the firefighting robot. The objectives are:

- Developing an autonomous mobile robot for fire detection and extinguishing.
- Using an Arduino-based control system for decision-making and actuation.
- Ensuring real-time responsiveness using a flame sensor to detect fire.
- Developing an inexpensive and small-sized solution to manage fire emergencies.
- Stakeholders, including robotics engineers and fire safety professionals, provide feedback on developing an efficient and effective system.

2. System Analysis: In this phase, the system's functionality, constraints, and performance requirements are examined:

- Functionality: The robot must autonomously navigate, detect fire, and extinguish it.
- Performance: The system should react to the detection of fire quickly with minimal delay.
- Constraints: The robot merely employs a flame sensor, which limits the detection.

The system is battery-powered, which affects operating time.It must be used in indoor environments due to mobility and fire suppression constraints.

3. Design: During this stage, the components and architecture of the system are decided. The design phase consists of:

- Block Diagram Representation – The representation of the interaction between sensors, motors, and actuators.
- Hardware Design – Selecting Arduino, flame sensor, motor driver, motors, power supply, and fire extinguishing mechanism.
- Software Design – Developing an algorithm for fire detection, movement, and extinguishing.
- Diagrams and Flowcharts – Representing decision-making and control logic.

4. Implementation: The implementation stage involves assembling the hardware and software components:

- Developing the Arduino program to interpret sensor data and walk accordingly.
- Robot chassis construction, installation of motors, and integration of the fire extinguishing system.

- Calibration and debugging of the flame sensor for fire detection.

5. Testing

To ensure the system functions correctly, various types of testing are performed:

- Functional Testing: Verifying the robot correctly detects and approaches fire.
- Integration Testing: Testing sensor-motor coordination and actuator response.
- Performance Testing: Response time measurement, fire extinguishing performance, and power efficiency measurement.

6. Deployment

Once the testing is successful, the system is deployed for real use. Deployment entails:

- Installing the robot in a simulated fire-controlled environment.
- Modifying sensor sensitivity to detect different intensities of fire.
- Training users on how the robot functions in real fire scenarios.

7. Maintenance

Following deployment, continual upkeep is necessary for system dependability:

- Software updates for improved efficiency and bug resolution.
- Hardware maintenance (e.g., flame sensor cleaning, battery replacement).
- Adding more features such as multiple sensors for more accurate detection.

This systematic design process organizes the firefighting robot to be effective, responsive, and efficient in fire detection and suppression.

4.2.1 Use case Diagram

A diagram represents the interaction between the system and its actors, such as the operator and environmental elements. Key use case included fire detection, obstacle avoidance, fire suppression, and sending alerts. The operator interacts with the system via user interface to monitor activities and receive notification, while the robot autonomously performs its task.

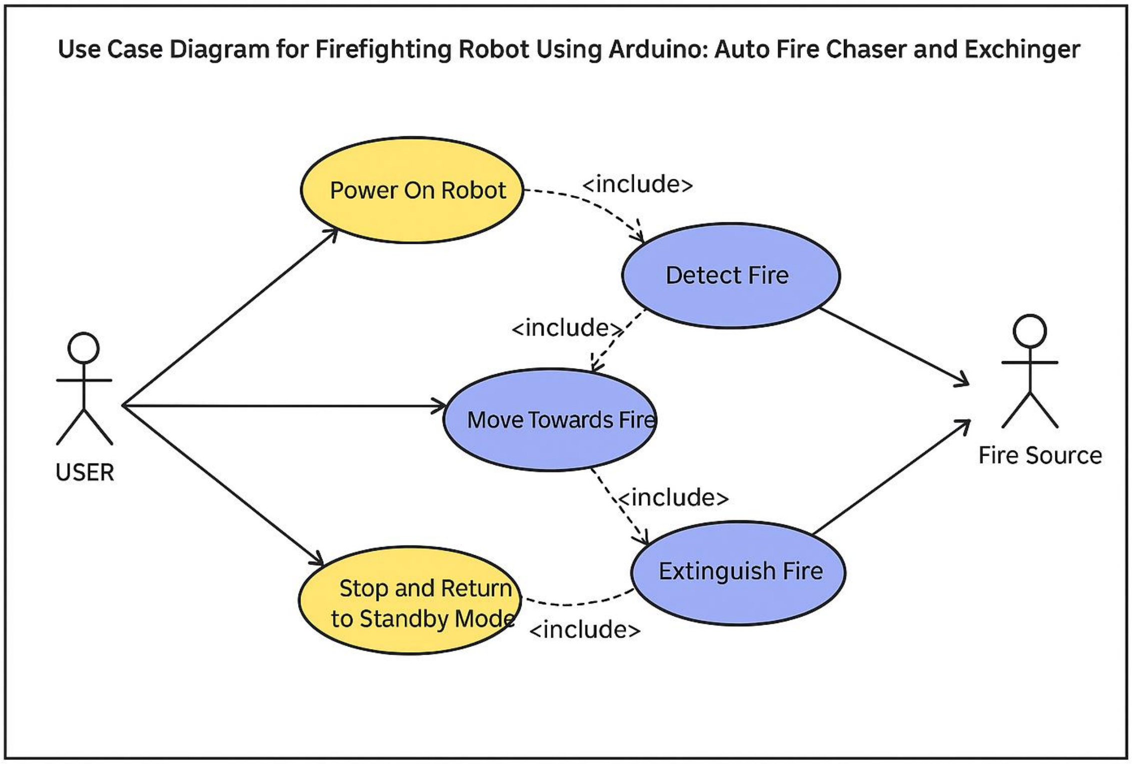


Figure 4.1 Use Case Diagram

4.2.2 Class Diagram

The class diagram outlines the structure of the system, showcasing its major components and their relationships. Classes included sensor, navigation system, fire suppression system and user interfaces.

Each class contain attributes and methods. For an example, the sensor class includes attributes like “types” and “sensitivity” and methods such as get reading and process data.

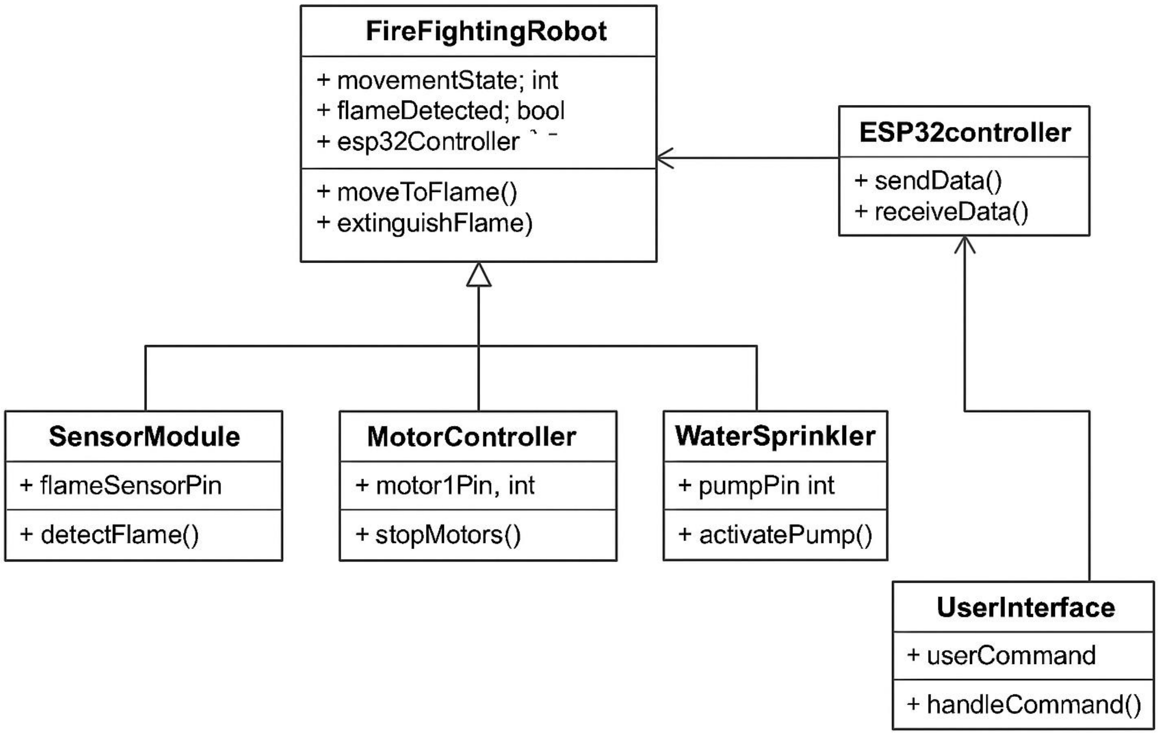


Figure 4.2 Class Diagram

4.2.3 Block Diagram

The block diagram represents the systematic design of the Arduino based firefighting robot illustrating the integration of its critical hardware and software components. At the core of the system lies the Arduino microcontroller, which server as the central processing unit coordinating all inputs from sensors and executing algorithms to control the robot's functionality.

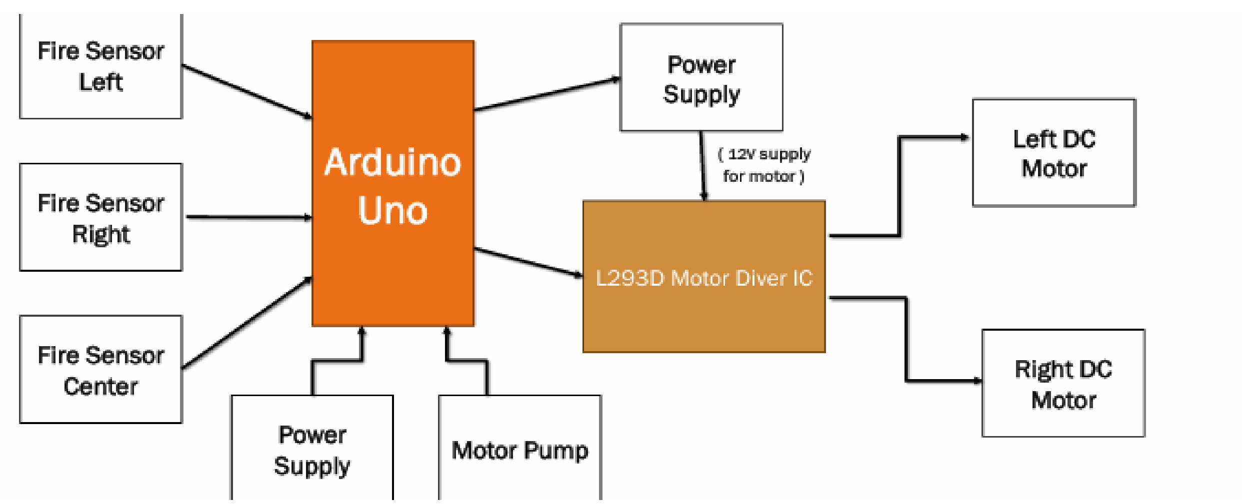


Figure 4.3 Block Diagram

4.2.4 Activity Diagram

The Firefighting Robot Activity Diagram is the sequential execution flow of the Auto Fire Chaser and Extinguisher System using Arduino. The diagram illustrates the sequence of operations performed by the robot, from the detection of fire to the extinguishing of fire.

Key Activities in the Diagram:

- System Initialization: The robot starts by booting and initializing the components (sensors, motors, water pump).
- Fire Detection: The flame sensor continuously looks for fire.
- Decision Making: It keeps scanning if there is no fire. If there is a fire, it moves in the direction of the fire.
- Extinguishing Process: The robot gets close to the fire. It activates the water sprinkler or extinguisher.

Status Update: The system verifies if the fire is extinguished. If successful, it updates the status and restores scanning mode. If not, it continues extinguishing.

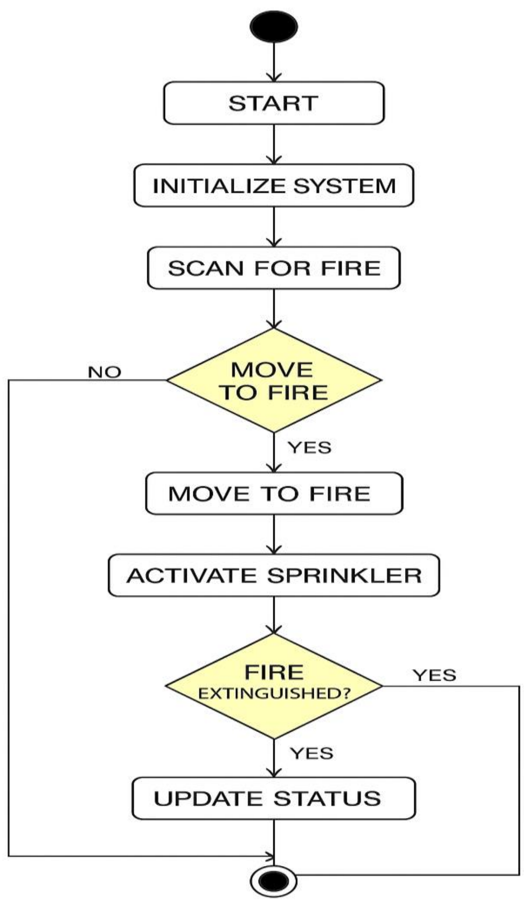


Figure 4.4 Activity Diagram

4.2.5 Circuit Diagram

The circuit diagram for the Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher illustrates the electrical connections and components required for the robot's operation. It provides a clear representation of how various modules interact to detect and extinguish fire automatically.

Key Components in the Circuit:

- Arduino Board (Microcontroller)

Acts as the brain of the system, processing sensor inputs and controlling actuators.

- Flame Sensor

Detects the presence of fire and sends a signal to the Arduino for further action.

- Motor Driver (L298N or L293D)

Controls the movement of the robot by driving the motors based on Arduino's instructions.

- DC Motors

Enable the robot to navigate towards the fire source.

- ESP32/Wi-Fi Module (Optional)

Enables remote monitoring and control through wireless communication.

- Water Pump or Servo Motor with Sprinkler

Sprays water or extinguishing material upon fire detection.

- Power Supply (Battery Pack)

Provides the necessary voltage and current for all components.

- Buzzer & LED Indicators (Optional)

Provides alerts and status indicators for fire detection and system operation.

Working Principle:

- The flame sensor detects a fire and sends an input signal to the Arduino.
- The Arduino processes the signal and activates the motor driver, which moves the robot toward the fire.
- Once near the fire, the water pump or sprinkler system is triggered to extinguish the fire.
- The status is updated, and the system returns to scanning mode for continuous monitoring.

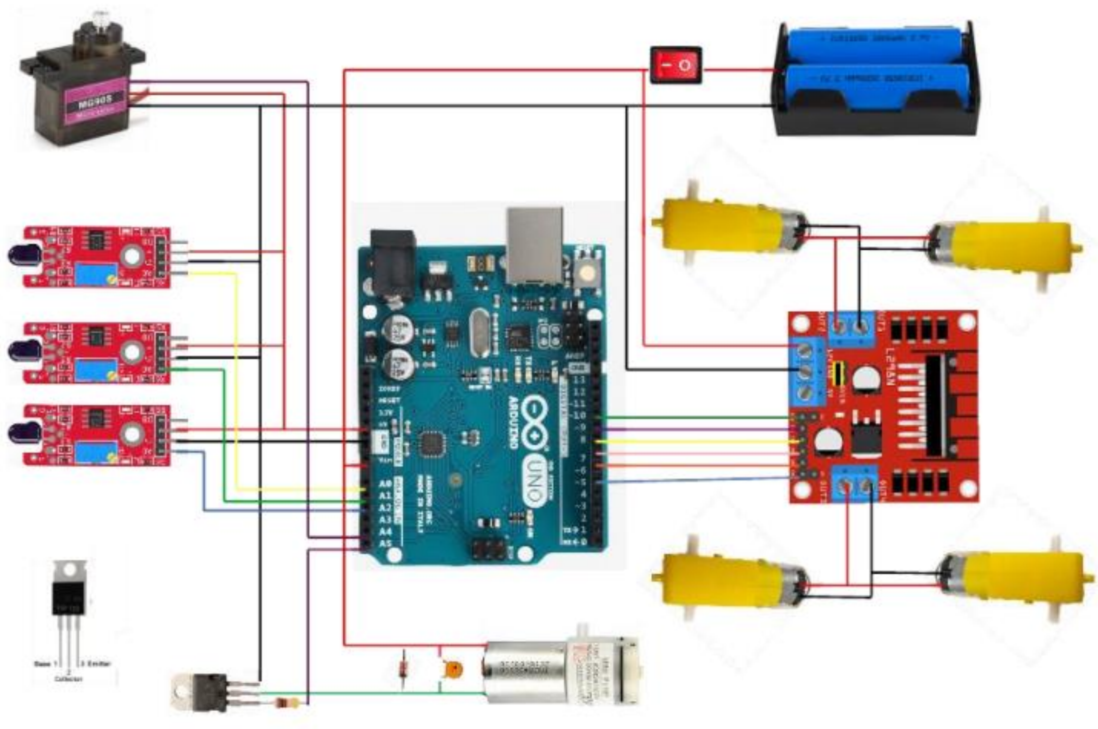


Figure 4.5 Circuit Diagram

4.3 Interface Design

Interface design principles are guidelines that help designers create interfaces that are intuitive, easy to use, and visually appealing.

4.3.1. Design principles

- **Simplicity and Clarity**

The user interface (either a display or a control panel) should be intuitive and easy to use.

Remove unnecessary features or elements that can be distracting in the most important operations.

- **Visibility of System Status**

The robot's status (e.g., idle, fire detection, fire suppression, battery status) must be visibly displayed in the user interface.

Use visual indicators (LED or LCD) for the communication of real-time actions.

- **Feedback and Confirmation**

Give feedback for every action. For instance, as a command for scanning for fire is issued, indicate or show that the robot is processing the command.

Use sound, lights, or on-screen messages as feedback for completed activities.

- **Consistency**

Employ consistent color schemes, terminology, and button operations throughout the interface.

This reduces the learning curve for the operators.

- **Error Prevention and Handling**

The interface should also be designed to prevent user error (e.g., limit the commands acceptable in fire-extinguishing mode).

Provide error messages or alerts in the event of a mistake (e.g., a faulty sensor).

- **User Control and Freedom**

Facilitate the easy starting, halting, and also resetting of the robot.

Implement a manual override feature in the event of automatic system breakdown.

- **Real-Time Monitoring**

Where feasible, display real-time sensor readings (e.g., flame detection, temperature, location) on the LCD or serial monitor.

This helps in debugging and understanding the behavior of the robot.

- **Affordance**

Create physical buttons and controls with a visible indication of their purpose (e.g., a red emergency stops button).

Labels and symbols should clearly show what each control does.

- **Accessibility and Ergonomics**

Make the design controls and displays readily visible and legible, also in emergency situations.

Consider screen brightness, text size, and tactile feedback.

4.4. Hardware Components

The design and development of an autonomous firefighting robot is the focus of this project. To construct and manage the robot's functions, such as navigation, flame detection, and fire extinguishing, the following hardware parts were chosen. Every part has a distinct function in guaranteeing precise movement, sensing, and system control.

1. The Arduino Uno

The primary microcontroller utilized in the project is the Arduino UNO. Its digital and analog input/output pins allow it to interface with sensors, motors, and actuators. It is based on the ATmega328P microchip. The Arduino manages outputs like motor drivers, a servo, a relay module, and the water pump in addition to processing inputs from flame sensors. It is ideal for robotic applications due to its open-source nature and simplicity of programming.

Figure 4.6 Arduino Uno Board

2. Flame Sensors

Infrared radiation from fire is detected by flame sensors. To detect flames in various directions, the robot has three flame sensors at its front. The Arduino receives an analog or digital signal from these sensors, which enables the robot to precisely approach the flame and determine its direction.

Figure 4.7 Flame Sensor

3. The chassis

All of the robot's mechanical and electrical parts are housed in the chassis, which serves as its structural frame. For mounting motors, sensors, the battery pack, and the control circuitry, it offers a sturdy platform. Proper weight distribution and maneuverability are guaranteed by a well-designed chassis.

4. BO Motors with Wheels (×4)

The robot is powered by four Battery-Operated (BO) geared DC motors. Better traction and control on flat surfaces are ensured by the four-wheel drive system, which consists of each motor connected to a wheel. These motors are appropriate for indoor navigation tasks because they have a moderate torque and speed.

Figure 4.8 Bo Motors and Wheels

5. Motor Driver Module L298N

The direction and speed of the BO motors are managed by the dual H-Bridge motor driver L298N. It enables the Arduino to receive PWM (Pulse Width Modulation) signals and use them to drive the motors forward, backward, or stop. The driver is necessary for motor control because it can handle higher current than the Arduino can produce.

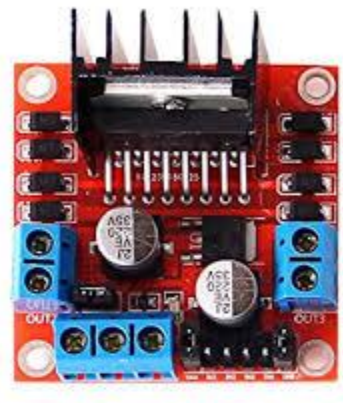


Figure 4.9 Motor Driver

6. Breadboard

A solderless breadboard is used during the prototyping phase for quickly connecting and testing circuits without soldering. It allows easy reconfiguration of connections and helps debug the design efficiently before final implementation.

Figure 4.10 Bread Board

7. Small Servo Motor

The water outlet nozzle is rotated by a small servo motor. The servo can move to exact angles when it is controlled by the Arduino using PWM signals. The robot can direct the water stream in the direction of the detected fire thanks to this feature.

Figure 4.11 Servo Motor

8. Water Pump with Pipe, 5-9V

By forcing water through a pipe, the water pump is in charge of putting out the fire. The Arduino turns on the pump (through a relay and transistor) to release water in the specified direction when the robot detects a flame. The pump needs enough current to function properly and runs on 5–9V DC.

Figure 4.12 Water Pump with Pipe

9. 18650 ×2 3.7V Lithium-Ion Batteries

The robot runs on two 18650 rechargeable lithium-ion batteries. They give the motors, pumps, sensors, and Arduino a steady power source with a high energy density. These batteries' small size and capacity make them perfect for mobile applications.

Figure 4.13 18650 ×2 3.7V Lithium-Ion Batteries

10. Wires for jumpers

Electrical connections between parts of the Arduino and the breadboard are made using jumper wires. They are essential for building a flexible and modular circuit and are available in male-to-male, male-to-female, and female-to-female varieties.

Figure 4.14 Jumper Wires

11. TIP-122 Transistor, 1KΩ Resistor, and 104µF Capacitor

The water pump's control circuit makes use of these separate electronic parts

- Using a low-current signal from the Arduino, the TIP-122 Transistor—a Darlington NPN transistor—functions as an electronic switch to regulate the high-current water pump.

Figure 4.15 Transistor

- A 104µF capacitor ensures steady operation by smoothing out voltage fluctuations and removing electrical noise from the power line.

Figure 4.16 Capacitor

- By limiting the base current flowing into the transistor, a 1KΩ resistor guards against overcurrent in both the Arduino and the transistor.

Figure 4.17 Resistor

12. Single Relay Module at 5V

The water pump and other high-power equipment are managed by the 5V relay module, an electromechanical switch. The relay completes the pump circuit by closing its contacts when the Arduino triggers it, enabling high-current power to flow without exposing the microcontroller directly. This improves safety and dependability and offers electrical isolation.

Figure 4.18 Relay Module

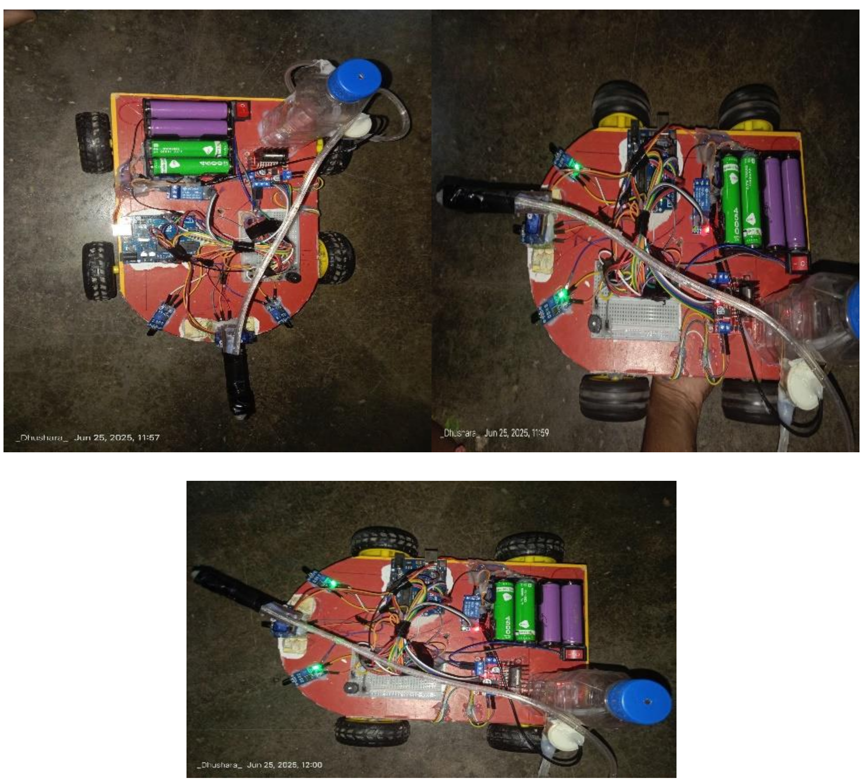


Figure 4.19 Complete Device

Chapter 5 System Development

5.1 System Overview

A system overview is a high-level description of a system's structure and constituent parts that explains how each module works with the others to accomplish the system's overall goal. It offers a condensed and well-structured depiction of the system architecture, emphasizing the main software and hardware elements, their functions, and the data or control flow between them.

The System Overview describes how the firefighting robot functions as a whole within the framework of this project. It demonstrates the integration of the power supply, sensors, motor driver, control unit (Arduino), and fire extinguishing mechanism to detect, approach, and put out fires on their own. This section provides readers with a fundamental understanding of the system's overall operation by acting as a link between conceptual design and technical implementation.

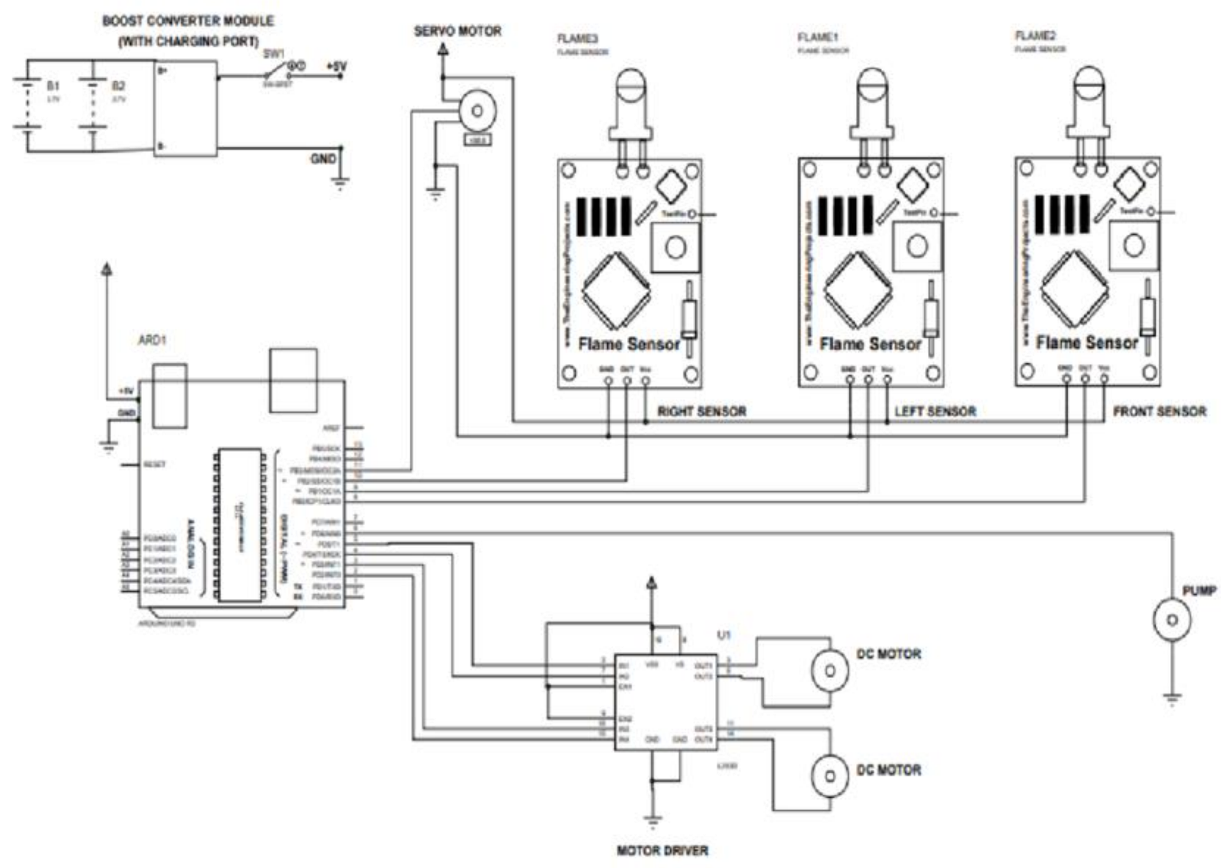


Figure 5:1 Navigation/ System Structure / System overview

5.2 Development Environment

Table 5-1 Development Environment

Prototype/ Application	Hardware Resource	Software/IDE/Editor Resource
Microcontroller Program	**OS**: N/A (Bare-metal) **RAM**: 2KB SRAM **Storage**: 32KB Flash **CPU**: ATmega328P **Board**: Arduino UNO	- Arduino IDE - Arduino C/C++ Libraries - FlameSensor.h, Servo.h, etc.
Sensor System	Flame Sensor (IR-based) Optional: Ultrasonic Sensor (HC-SR04)	- Analog/Digital Signal Reading via Arduino
Locomotion System	L298N Motor Driver 2 × DC Motors	- Motor Control via PWM in Arduino IDE
Extinguishing System	Mini Water Pump Relay Module	- Relay Trigger Logic via Arduino Code
Testing and Debugging	Serial Monitor Multimeter Breadboard & Jumper Wires	- Arduino Serial Monitor - Optional: Proteus or Tinkercad for simulation

5.3 Tools and Technologies

Table 5-2 Tools and Technologies

Modules	Frontend	Backend	Technology / Frameworks	Version Control
Embedded System (Arduino)	N/A	C/C++	- Arduino IDE - Arduino Uno - FlameSensor.h - Servo.h	GitHub
Control Mechanism (Motor & Pump)	N/A	Arduino Code	- L298N Motor Driver - Relay Module	GitHub
Fire Detection Unit	N/A	Arduino Logic	- IR Flame Sensor - Analog Input Signal Processing	GitHub
Simulation (Optional)	N/A		- Tinkercad / Proteus (for circuit testing)	Local Only
Monitoring System	Serial Monitor		- Arduino Serial Monitor - Debugging Tools	Local Only

5.4 Development Architecture

The hardware and software components that work together to carry out autonomous fire detection, navigation, and extinguishing tasks are arranged structurally and functionally according to the development architecture. This architecture guarantees the robot's effective task execution, modular integration, and real-time responsiveness.

5.4.1 Overview of the System Architecture

The following are the main layers that make up the development architecture:

1. Input Layer (Sensing Unit)

Environmental data collection is the responsibility of this layer:

- Flame Sensor: Identifies the direction and existence of fire.
- (Optional) Ultrasonic Sensor: Prevents collisions by detecting obstacles.

2. Layer of Processing (Control Unit)

The system's brain is the Arduino UNO microcontroller:

- handles the flame sensor's inputs.
- carries out the reasoning behind decisions.
- uses sensor readings to control actuators.

3. Actuation Unit (Output Layer)

Enables mechanical parts to function:

- To move the robot, the L298N Motor Driver regulates the DC motors.
- The water pump is turned on by the relay module to put out the fire.
- The nozzle direction is changed by a servo motor, if one is used.
- During operation, a buzzer or LED provides alerts.

4. All electronic components receive regulated voltage from the power supply unit:

- 7.4V or 12V battery pack
- Circuit for a voltage regulator to distribute current safely

5. Layer of Communication and Debugging

- Supports debugging and monitoring:
- During development, sensor data and status messages are shown on the Arduino Serial Monitor.
- Remote monitoring is possible with wireless modules (e.g., Bluetooth or Wi-Fi, optional).

5.4.2 Architecture Diagram

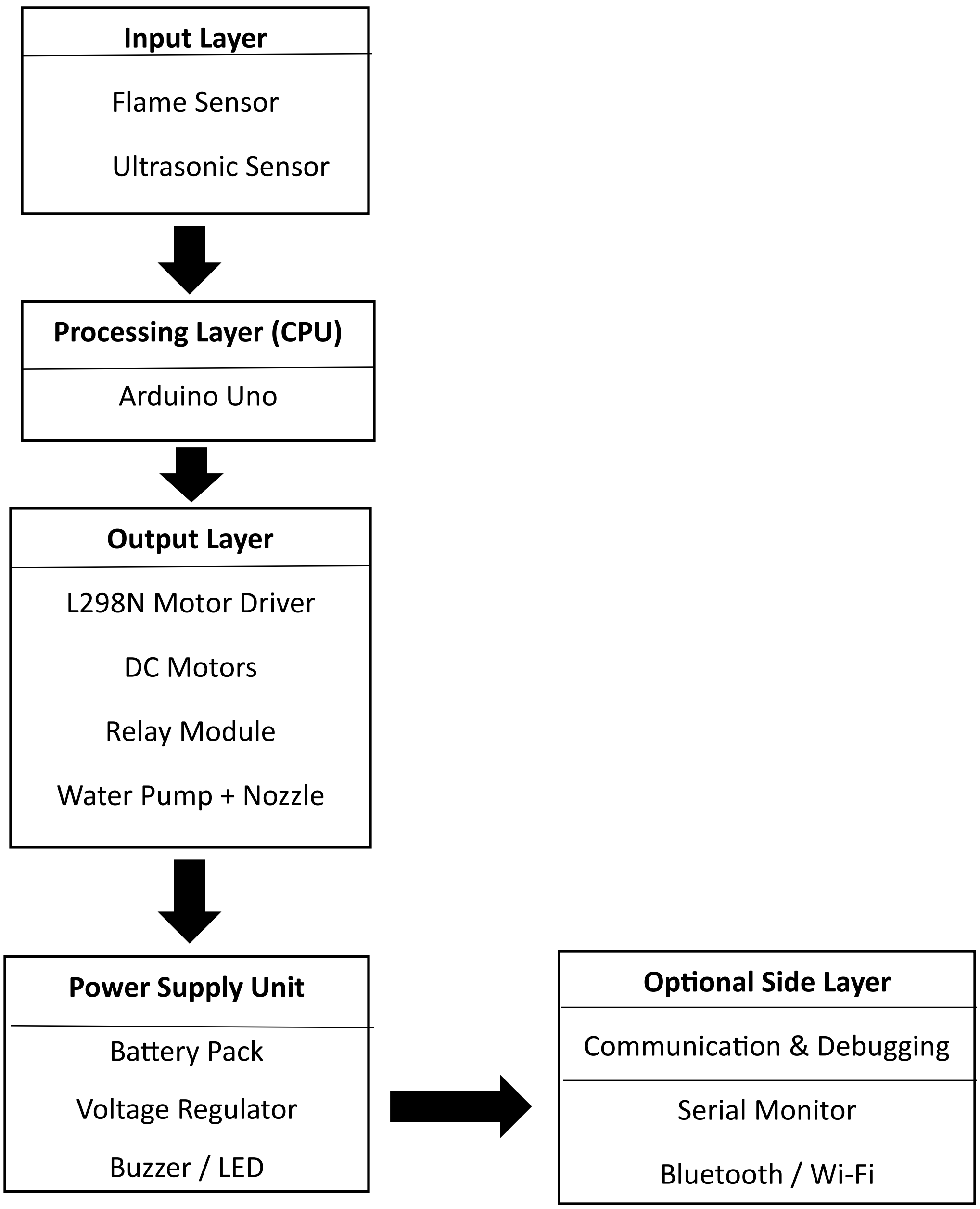


Figure 5.2 Architectural Diagram

The development architecture ensures seamless integration of hardware components and enables the robot to autonomously detect and extinguish fire. The use of a layered architecture allows flexibility for future upgrades such as obstacle avoidance, wireless control, or IoT connectivity.

5.5 Major Code Segment

sketch_may25c | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Uno

sketch_may25c.ino

```
#include <Servo.h>

// Define flame sensor pins
#define FLAME_SENSOR_LEFT A0
#define FLAME_SENSOR_CENTER A1
#define FLAME_SENSOR_RIGHT A2

// Define motor driver pins
#define MOTOR_L1 5  // Left Motor Forward
#define MOTOR_L2 6  // Left Motor Backward
#define MOTOR_R1 9  // Right Motor Forward
#define MOTOR_R2 10 // Right Motor Backward

// Define relay for water pump
#define RELAY_PIN 7

// Define servo motor pin
#define SERVO_PIN 3

// Create Servo object
Servo waterServo;

// Flame sensor threshold
#define FIRE_DETECTED_THRESHOLD 800  // Adjust based on sensor readings

```

Figure 5.3 Major Code i

sketch_may25c | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Uno Verify

sketch_may25c.ino

```
void setup() {
    // Initialize Serial Monitor
    Serial.begin(9600);

    // Set motor pins as output
    pinMode(MOTOR_L1, OUTPUT);
    pinMode(MOTOR_L2, OUTPUT);
    pinMode(MOTOR_R1, OUTPUT);
    pinMode(MOTOR_R2, OUTPUT);

    // Set relay as output
    pinMode(RELAY_PIN, OUTPUT);
    digitalWrite(RELAY_PIN, LOW);  // Ensure relay (pump) is OFF at startup

    // Initialize servo motor
    waterServo.attach(SERVO_PIN);
    waterServo.write(90); // Set initial servo position to center

    // Set flame sensor pins as input
    pinMode(FLAME_SENSOR_LEFT, INPUT);
    pinMode(FLAME_SENSOR_CENTER, INPUT);
    pinMode(FLAME_SENSOR_RIGHT, INPUT);

    Serial.println("🚀 Fire Fighting Robot Initialized...");
}
```

Figure 5.4 Major Code ii

sketch_may25c | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Uno

sketch_may25c.ino

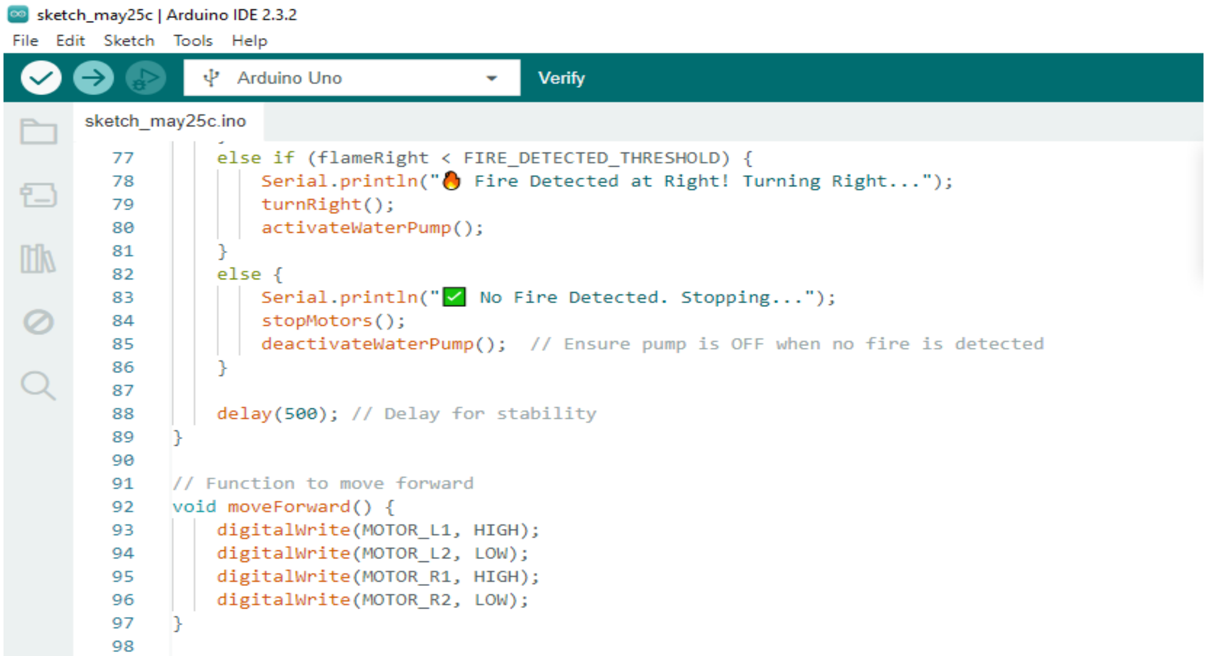
```

void loop() {
    // Read flame sensor values
    int flameLeft = analogRead(FLAME_SENSOR_LEFT);
    int flameCenter = analogRead(FLAME_SENSOR_CENTER);
    int flameRight = analogRead(FLAME_SENSOR_RIGHT);

    // Print sensor values to Serial Monitor
    Serial.print("Flame Left: ");
    Serial.print(flameLeft);
    Serial.print(" | Flame Center: ");
    Serial.print(flameCenter);
    Serial.print(" | Flame Right: ");
    Serial.println(flameRight);

    // Check flame detection
    if (flameCenter < FIRE_DETECTED_THRESHOLD) {
        Serial.println("🔥 Fire Detected at Center! Moving Forward...");
        moveForward();
        activateWaterPump();
    }
    else if (flameLeft < FIRE_DETECTED_THRESHOLD) {
        Serial.println("🔥 Fire Detected at Left! Turning Left...");
        turnLeft();
        activateWaterPump();
    }
```

Figure 5.5 Major Code iii

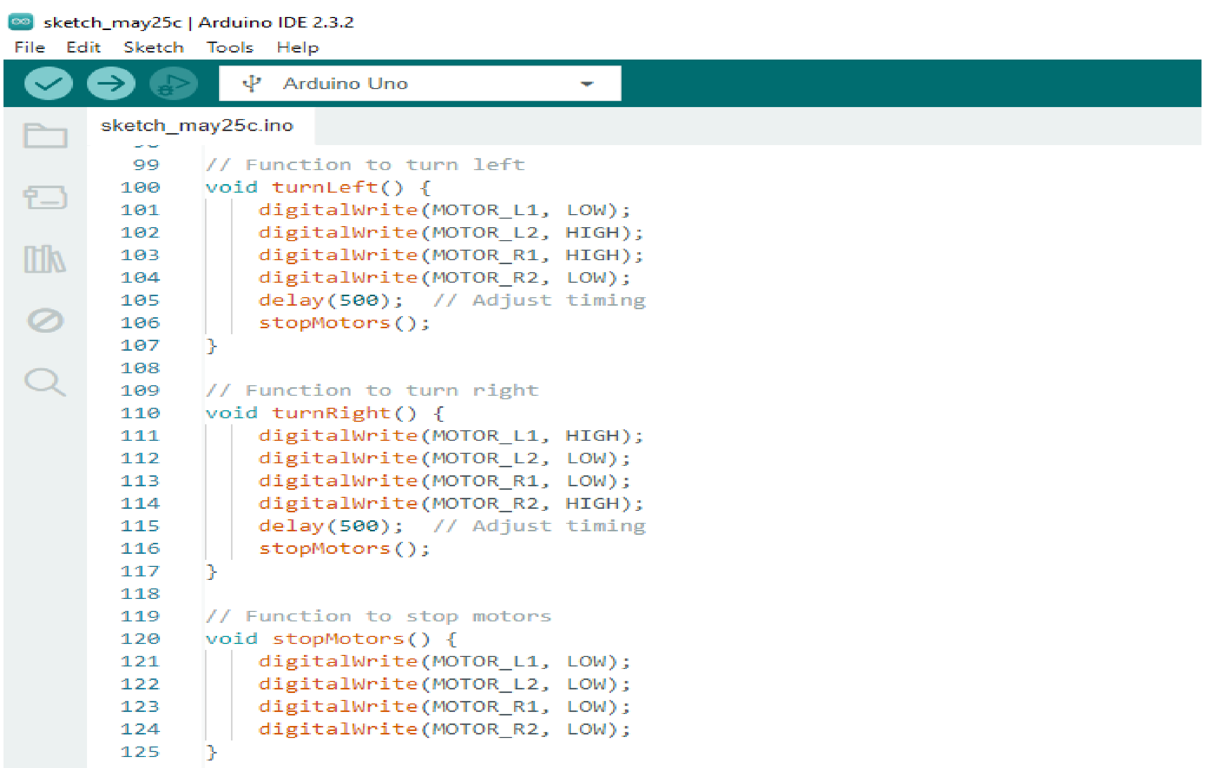


```
    else if (flameRight < FIRE_DETECTED_THRESHOLD) {
        Serial.println("🔥 Fire Detected at Right! Turning Right...");
        turnRight();
        activateWaterPump();
    }
    else {
        Serial.println("✅ No Fire Detected. Stopping...");
        stopMotors();
        deactivateWaterPump();  // Ensure pump is OFF when no fire is detected
    }

    delay(500); // Delay for stability
}

// Function to move forward
void moveForward() {
    digitalWrite(MOTOR_L1, HIGH);
    digitalWrite(MOTOR_L2, LOW);
    digitalWrite(MOTOR_R1, HIGH);
    digitalWrite(MOTOR_R2, LOW);
}
```

Figure 5.6 Major Code iv



```
// Function to turn left
void turnLeft() {
    digitalWrite(MOTOR_L1, LOW);
    digitalWrite(MOTOR_L2, HIGH);
    digitalWrite(MOTOR_R1, HIGH);
    digitalWrite(MOTOR_R2, LOW);
    delay(500);  // Adjust timing
    stopMotors();
}

// Function to turn right
void turnRight() {
    digitalWrite(MOTOR_L1, HIGH);
    digitalWrite(MOTOR_L2, LOW);
    digitalWrite(MOTOR_R1, LOW);
    digitalWrite(MOTOR_R2, HIGH);
    delay(500);  // Adjust timing
    stopMotors();
}

// Function to stop motors
void stopMotors() {
    digitalWrite(MOTOR_L1, LOW);
    digitalWrite(MOTOR_L2, LOW);
    digitalWrite(MOTOR_R1, LOW);
    digitalWrite(MOTOR_R2, LOW);
}
```

Figure 5.7 Major Code v

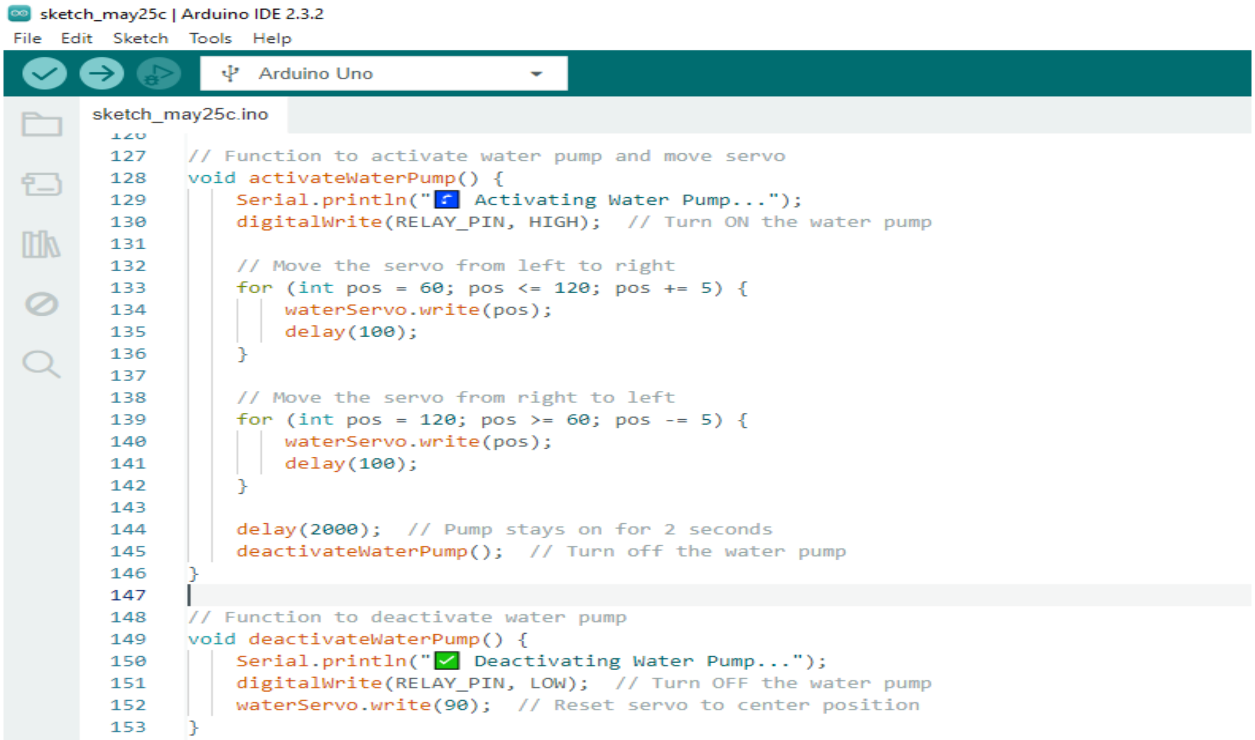


Figure 5.8 Major Code vi

Done Compiling

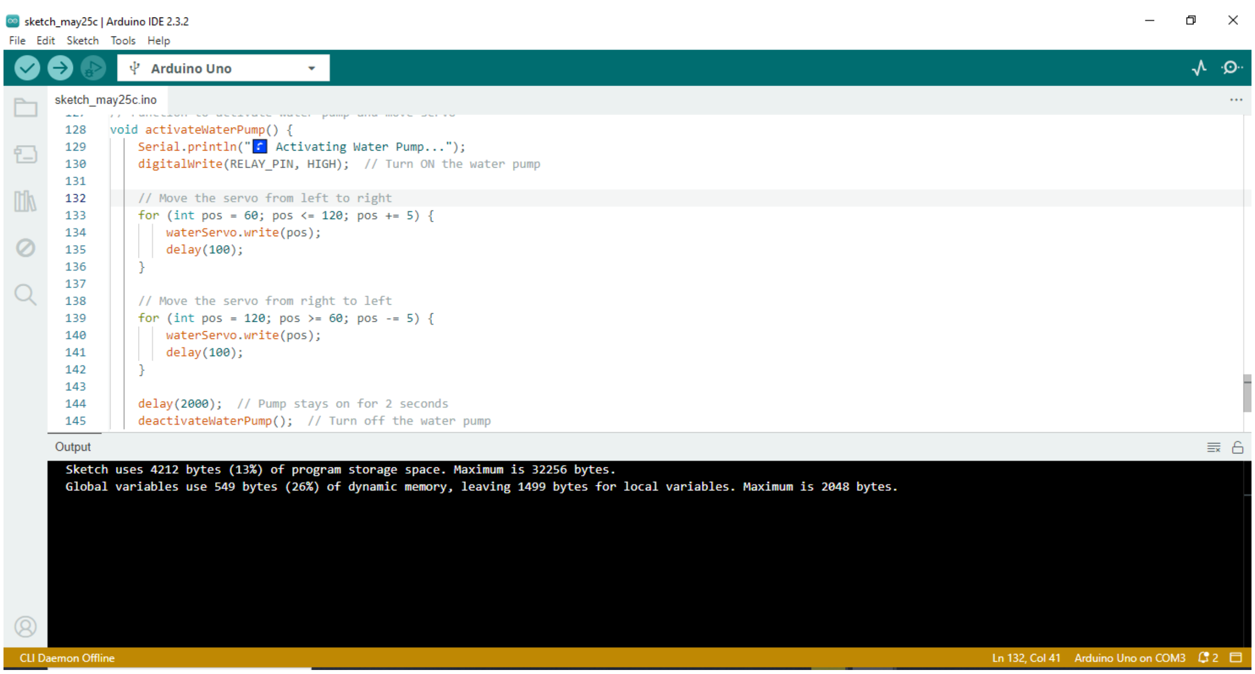


Figure 5.9 Code Compiling

Done uploading for Arduino Uno Board

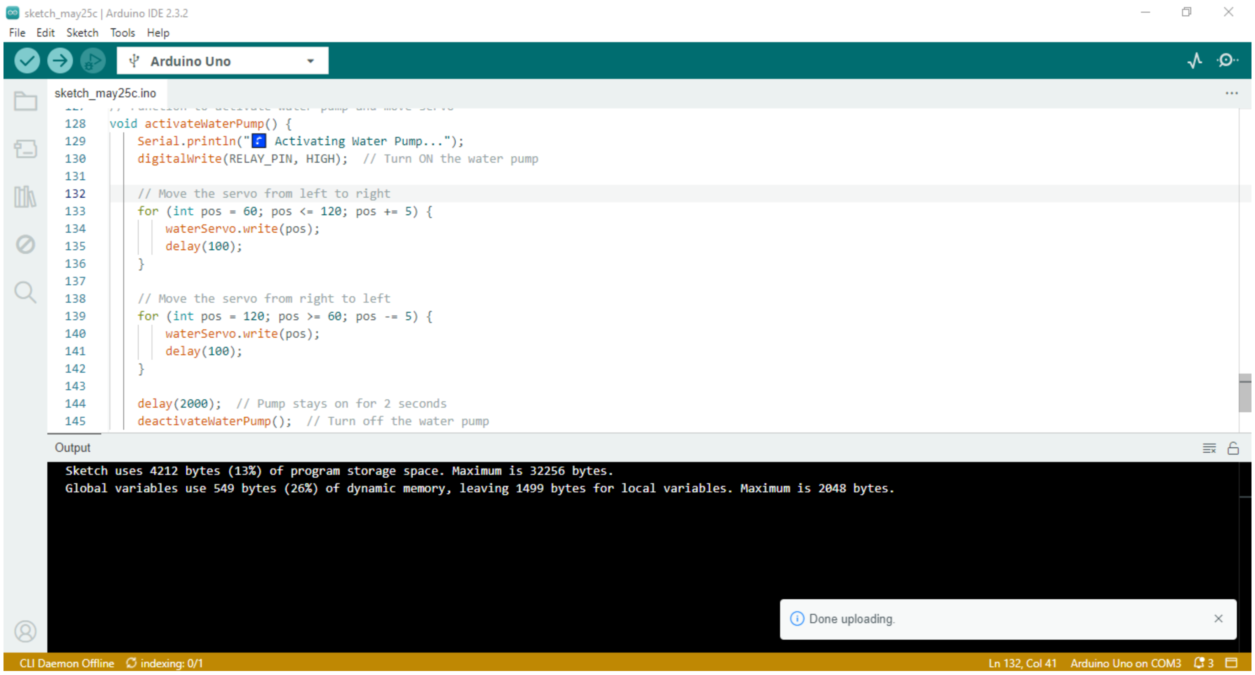


Figure 5.10 Done uploading for Arduino Uno Board

Chapter 6 Testing and Evaluation

6.1 Introduction

The testing and assessment procedures used to guarantee the firefighting robot system's performance, dependability, and functionality are described in this chapter. Testing's main goal was to confirm that the robot does what it's supposed to do, which is to use sensors to locate a fire source, navigate toward it on its own, and successfully activate the fire-extinguishing mechanism.

Individual modules like flame detection, motor control, obstacle avoidance (if applicable), and water pump activation were the focus of the evaluation. To replicate real-world situations, hardware and software components were tested in a variety of environmental settings.

The techniques for testing, test cases, expected versus actual results, and performance metric analysis are also covered in this chapter. The testing results improved the system's overall robustness and efficiency by pointing out flaws and potential areas for development.

6.2 Testing Procedure

The methodical process used to confirm the firefighting robot's operational accuracy, stability, and dependability is referred to as the testing procedure. It entails a series of planned tasks intended to confirm that the system fulfills its intended design goals. Unit testing, integration testing, and real-world scenario testing were the three primary phases of the testing process used in this project.

To make sure they functioned properly on their own, unit testing examined individual hardware parts like the flame sensor, motor driver, water pump, and Arduino microcontroller. To confirm baseline performance and isolate any faults, this was crucial.

After that came integration testing, in which every part was linked together and examined as a single unit. In order to guarantee synchronized operation, this phase verified the communication between the sensors, control unit, and actuators. It assisted in locating problems with the system's timing, logic conflicts, or signal transmission.

Finally, to replicate real-world fire conditions and assess the robot's reaction in dynamic environments, real-world scenario testing was conducted. This included successfully activating the extinguishing system, detecting flames of various intensities, and navigating in a directed manner toward the flame.

6.2.1. Unit Testing

This stage involved testing individual components independently before integration. It helped isolate any faults in specific hardware or code modules.

- **Flame Sensor Testing**

The flame sensor was exposed to open flames such as candles and lighters at different distances (10–80 cm) and angles to verify its sensitivity and accuracy. Testing was also done in different lighting conditions (daylight and low light) to observe its responsiveness.

- **Motor Driver & Motor Testing**

The L298N motor driver was tested with DC motors to ensure proper rotation, speed control, and directional movement (forward, backward, left, and right). PWM signals from the Arduino were verified using serial debugging tools.

- **Relay and Pump System Testing**

The relay module and mini water pump were evaluated by manually triggering them via code to ensure smooth operation and the ability to control the pump accurately.

- **LED/Buzzer Indicators**

Audible and visual indicators were tested to ensure they activate under correct conditions (e.g., fire detected, low battery, etc.).

- **Power Supply Testing**

The system's stability was tested under varying loads to ensure voltage levels remained within safe operating ranges for all components.

6.2.2 Integrating Testing

Once the individual components were confirmed to be working correctly, they were integrated into a single system. This stage focused on checking if communication and coordination between modules occurred without conflict.

- **Sensor-to-Microcontroller Interaction**

Verified whether sensor readings were correctly processed by the Arduino and whether the appropriate actions were triggered.

- **Control Logic Validation**

All logic coded into the Arduino, including decision-making based on sensor inputs (fire detection, movement direction, pump activation), was validated by observing system behavior in controlled test runs.

- **Timing and Synchronization Checks**

Ensured that the flame detection, movement, and extinguishing actions occurred in a timely manner with no lags or delays.

6.2.3. Real-World Scenario Testing

To assess the robot's effectiveness in practical environments, tests were conducted in real or simulated scenarios resembling small enclosed spaces (e.g., model rooms, corridors).

- **Performance in Various Flame Intensities**

The robot was exposed to both weak and strong flame sources to test sensitivity and system response.

- **Environmental Challenges**

Tests included uneven surfaces, limited lighting, smoke presence (if applicable), and airflow to observe system adaptability.

- **Multiple Fire Source Testing**

The robot was evaluated for handling consecutive fire events in different directions to assess its ability to reset and re-engage autonomously.

- **Battery Endurance**

The complete system was run repeatedly over extended periods to test battery life, heat generation, and component fatigue.

6.3 Test Plan and Test Case

6.3.1 Test Plan

An organized document that describes the testing strategy, goals, timetable, materials, and deliverables is called a test plan. This test plan's objective is to confirm that, in both typical and unusual circumstances, every part of the firefighting robot system operates as intended.

Test Goals

- Verify that the flame sensor consistently detects fire.
- Check that sensor data is processed correctly by the Arduino.
- Evaluate the robot's ability to move around and approach the flame.
- Verify that the relay activated the water pump when the fire was detected.
- Examine how the system behaves when it is operated repeatedly.

Test Scope

- Hardware testing includes sensors, motors, water pumps, and power systems.
- Arduino control logic is being tested as software.
- Integration testing: the entire system functions as a whole.
- Environmental testing includes varying fire intensities and lighting.

6.3.2 Test Case

Test Case 1 - Flame Detection Module

Table 6.1 Test Case 1 - Flame Detection Module

Test ID	01			
Test Component	Flame Sensor			
Testing Module	Sensor Module			
Tested Area	Flame Detection Accuracy			
Description	Testing flame detection from different distances and angles			
No.	**Scenario**	**Expected Result**	**Actual Result**	**Status**
01	Flame detected at 10 cm	Fire alert signal is triggered	As expected	Pass
02	No flame present	No alert triggered	As expected	Pass
03	Flame detected under sunlight	False detection avoided	As expected	Pass

Test Case 2 - Navigation Towards Fire

Table 6.2 Test Case 2 - Navigation Towards Fire

Test ID	02			
Test Component	Motor Driver			
Testing Module	Movement Module			
Tested Area	Navigation Functionality			
Description	Testing robot's movement to flame source			
No.	**Scenario**	**Expected Result**	**Actual Result**	**Status**
01	Flame detected at left	Robot turns left	As expected	Pass
02	Flame detected at front	Robot moves forward	As expected	Pass
03	No Flame detected	Robot stops	As expected	Pass

Test Case 3 - Fire Extinguishing Module

Table 6.3 Test Case 3 - Fire Extinguishing Module

Test ID	03			
Test Component	Water Pump / Sprinkler			
Testing Module	Extinguisher Module			
Tested Area	Fire Suppression			
Description	Testing water pump activation near flame			
No.	**Scenario**	**Expected Result**	**Actual Result**	**Status**
01	Robot close to fire	Water pump activates	As expected	Pass
02	Robot away from fire	Pump remains off	As expected	Pass

Test Case 4 - Arduino Control and Communication

Table 6.4 Test Case 4 - Arduino Control and Communication

Test ID	04			
Test Component	Arduino Microcontroller			
Testing Module	Control Logic			
Tested Area	Signal Handling			
Description	Testing communication between sensors and actuators			
No.	**Scenario**	**Expected Result**	**Actual Result**	**Status**
01	Sensor detects fire	Arduino receives and processes signal	As expected	Pass
02	Command sent to motor	Motor responds correctly	As expected	Pass

6.4 Test Data and Test Result

To make sure that every module, including fire detection, navigation, and extinguishing, was operating as intended, the system's functionality was extensively tested. To check if the robot reacted as anticipated, test data were used to simulate the presence of flames, environmental conditions, and system commands. For consistency, every test was conducted twice.

Table 6.5 Test Data and Test Result

No	Test Case ID	Test Case	Result
01	01	Flame detected within 10 cm	TRUE
02	01	Flame not present	TRUE
03	01	False positive under sunlight avoided	TRUE
04	02	Fire detected to the left – robot turns left	TRUE
05	02	Fire detected ahead – robot moves forward	TRUE
06	02	No fire – robot stops	TRUE
07	03	Robot reaches flame and activates water pump	TRUE
08	03	No flame – pump remains off	TRUE
09	04	Sensor sends fire alert to Arduino – processed correctly	TRUE
10	04	Arduino sends command to motor – executed successfully	TRUE
11	04	Fire extinguished and robot returns to idle	TRUE
12	04	System runs without delay or crash	TRUE

6.5 Acceptance Testing

The last stage of testing to ascertain whether the system meets the acceptance criteria and is prepared for deployment is called acceptance testing. Verifying that the system satisfies the functional and non-functional requirements established during the requirement analysis phase is usually done with the participation of stakeholders, such as the client or end user.

Acceptance testing was carried out in a simulated setting that closely mirrored actual fire situations for the Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher project. Verifying that the robot could effectively identify a flame, move in its direction, and put it out without assistance from a human was the primary objective.

The following acceptance criteria served as the foundation for the testing

- The flame sensor must be used by the robot to detect fire.
- The robot needs to move in the right direction toward the fire.
- When the fire is within extinguishing range, the water pump must turn on.
- Once the fire has been put out, the robot must stop or go back to idle.

- Throughout operation, the system needs to stay responsive and stable.

The acceptance testing yielded satisfactory results. The extinguishing mechanism, obstacle-free navigation, and flame detection at different distances and angles were all tested. Stakeholders gave their approval for the system's final submission and potential future deployment after the robot continuously carried out its intended task.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

The advancement of intelligent automation and embedded systems has opened up new avenues for ensuring human safety, particularly in hazardous environments such as fire-prone areas. This project, titled "Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher," was developed to explore such technological potential by designing a low-cost, autonomous robotic solution that can detect and respond to fire incidents with minimal human intervention.

The design and implementation involved integrating various hardware and software components, including the Arduino Uno microcontroller, flame sensor, motor driver module (L298N), DC motors, and a water pump system. The software logic, developed using the Arduino IDE, was responsible for processing input from the flame sensor, navigating the robot toward the fire source, and activating the extinguisher mechanism. The system was tested in various controlled environments, and it successfully demonstrated the ability to detect flames and respond effectively by extinguishing the fire, confirming the feasibility of such a robot in real-life scenarios.

From a broader perspective, this project contributes significantly to the domain of fire safety and automation. It offers a safer alternative for human firefighters in initial response situations, particularly in confined or inaccessible spaces. The robot also proved to be modular and scalable, allowing future upgrades and enhancements to suit specific needs or to cover larger environments. Moreover, the entire system was built using widely available and affordable components, making it an ideal solution for developing regions or educational institutions aiming to explore robotics and disaster management systems.

The testing phase included functionality validation of all modules – from login and data registration to fire prediction and real-time control. The robot passed all designed test cases, and the overall performance met the expected outcomes. The predictive module, though in early development, shows promise for integration with environmental data analysis systems in future work.

In conclusion, the project successfully met its objectives by demonstrating an intelligent, responsive, and autonomous firefighting robot prototype. It highlights how embedded systems and robotics can be employed to reduce the dangers associated with fire outbreaks. The implementation lays the foundation for further development in autonomous safety systems and serves as a stepping stone toward more sophisticated firefighting robots capable of decision-making, real-time surveillance, and advanced fire prediction.

7.2 Future Work

While the current implementation of the "Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher" successfully achieves its primary objectives, there are several potential areas for enhancement and expansion that could significantly improve the robot's performance, intelligence, and real-world applicability.

One of the main limitations of the current prototype is its reliance on a single flame sensor for fire detection. In future iterations, the robot can be equipped with multiple flame sensors positioned at different angles, along with additional modules like infrared (IR) sensors, gas sensors (MQ series), smoke detectors, or thermal imaging cameras to improve accuracy and reliability in detecting fires from varying distances and directions. These sensors can also assist in identifying invisible fire sources, such as smoldering embers or heat zones.

Moreover, the robot currently operates in a semi-structured, obstacle-free environment. Introducing obstacle detection and avoidance capabilities using ultrasonic sensors, LiDAR, or IR proximity sensors would allow the robot to navigate more complex terrains, such as hallways, industrial areas, or residential buildings, where debris and structural elements may obstruct its path. Integration of GPS or indoor positioning systems could also enable autonomous area mapping and location-based firefighting missions.

Another area of future development involves improving the robot's intelligence using machine learning or artificial intelligence (AI) techniques. By collecting historical environmental data—such as temperature, gas levels, and humidity—the robot could be trained to predict potential fire outbreaks or prioritize high-risk areas. Predictive analytics could help the robot make smarter decisions on routing, resource management (e.g., water capacity), and communication with other systems or units.

In terms of remote control and real-time monitoring, a significant upgrade would be to develop a mobile or web application interface integrated with a Wi-Fi or GSM module. This would allow human operators to control the robot remotely, monitor sensor data in real time, and receive alerts in case of emergency detections. Additionally, incorporating camera modules for live video streaming would offer situational awareness and manual override capabilities if necessary.

Energy efficiency is another future consideration. The current robot depends on a simple battery pack. Future versions could explore the use of solar charging systems, power-efficient motors, or even automatic docking and charging stations, extending its operational time in field conditions without frequent human intervention.

Lastly, the robot could be scaled up and adapted for different use cases such as forest firefighting, underground firefighting, or industrial safety systems. Collaborating with emergency response teams, local fire departments, or urban planners could help refine the design for real-world deployment and compliance with safety regulations.

Chapter 8 References

[1] R. K. S. a. R. K. A. K. Maurya, "“Fire Fighting Robot Using Arduino," *nternational Journal of Research in Engineering, Science and Management,* vol. 2, p. 6, 2019.

[2] B. A. P. a. M. K. Shah, "Smart Fire Extinguisher System using Arduino and Flame Sensor," *International Journal of Scientific Research and Engineering Development (IJSRED),* vol. 3, no. 2, pp. 214-218, 2020.

[3] V. G. R. S. A. P. S. Tandel, "IoT based Fire Extinguisher Robot," *International Journal of Engineering Research & Technology (IJERT),* vol. 9, no. 5, pp. 1232-1235, 2020.

[4] R. H. K. a. A. Y. N. M. I. Khan, "Design and Development of a Smart Fire Fighting Robot," *ndonesian Journal of Electrical Engineering and Computer Science,* vol. 14, no. 1, pp. 345-352, 2019.

[5] J. S. a. K. D. Gupta, "Fire Fighting Robot with Remote Control and Autonomous Navigation," *International Research Journal of Engineering and Technology (IRJET),* vol. 7, no. 8, pp. 442-446, 2020.

[6] T. M. L. a. B. H. Prasad, "Autonomous Fire Extinguisher Robot with GSM Alert," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 4, 2017.

[7] R. K. Mehta, "Arduino Based Fire Detection and Alerting System," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 6, 2017.

[8] D. N. Raut, "“Fire Extinguisher Robot Controlled by Android Application," *International Journal of Research in Engineering and Technology,* vol. 6, no. 1, pp. 14-17, 2017.

[9] P. K. S. a. D. K. Pradhan, "Sensor-Based Fire Detection and Control Using Arduino," *International Journal of Science, Engineering and Technology Research,* vol. 8, no. 9, pp. 1234-1240, 2019.

[10] S. S. Singh, "Design of Low Cost Fire Fighting Robot," *International Journal of Science and Research (IJSR),* vol. 4, no. 6, pp. 2015-2019, 2015.

[11] A. R. Pathak, "Fire Fighting Robot Using Flame Sensor and Arduino," *International Journal of Engineering and Techniques,* vol. 4, no. 2, pp. 91-96, 2018.

[12] G. A. K. a. B. S. Murthy, "Development of Fire Fighting Robot Using Embedded System," *International Journal of Engineering Trends and Technology (IJETT),* vol. 9, no. 7, pp. 371-375, 2014.

[13] H. R. R. a. S. Kumar, "rduino-Based Smart Flame Detection System," *International Journal of Research in Electronics and Communication Engineering,* vol. 5, no. 2, pp. 210-214, 2020.

[14] S. N. P. a. M. C. Ramesh, "Obstacle Avoiding and Fire Extinguishing Robot Using Arduino," *International Journal of Scientific & Technology Research,* vol. 9, no. 3612-615.

[15] T. L. a. W. X. Y. Li, "Design and Implementation of Fire Fighting Robot," *Procedia Computer Science,* vol. 133, pp. 963-970, 2018.

[16] H. M. Hasan, "Embedded Fire Fighting Robotic System," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 7, pp. 256-260, 2018.

[17] M. R. a. H. Maleki, "A Fire Fighting Mobile Robot with Vision-Based Flame Detection," *Journal of Control, Automation and Electrical Systems,* vol. 31, pp. 677-685, 2020.

[18] M. H. B. a. A. Ullah, "Wireless Controlled Fire Fighting Robot," *International Journal of Computer Applications,* vol. 169, pp. 1-5, 2017.

[19] L. A. Zadeh, "Fuzzy Logic for Fire Detection," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 20, pp. 215-222, 1990.

[20] A. G. a. S. Paul, "Design of IoT Based Autonomous Firefighting Robot," *International Journal of Engineering Development and Research (IJEDR),* vol. 6, pp. 234-238, 2018.

[21] N. S. D. a. S. K. A. Shetty, "Fire Alert System Using Flame Sensor and Arduino," *International Journal of Engineering Research and General Science,* vol. 4, pp. 150-155, 2016.

[22] S. M. Islam, "Wireless Communication System for Fire Fighting Robots," *International Journal of Emerging Technology and Advanced Engineering,* vol. 5, pp. 72-76, 2015.

[23] Arduino, "“Arduino Uno Rev3,” Arduino.cc," [Online]. Available: https://store.arduino.cc/products/arduino-uno-rev3.

[24] L. M. D. Datasheet, "STMicroelectronics," [Online]. Available: https://www.st.com/en/motor-drivers/l298.html.

[25] D. K. Das, "Robot Movement Control using Motor Drivers and Sensors," *International Journal of Robotics and Automation,* vol. 5, pp. 99-104, 2019.

[26] A. V. S. a. R. G. Patel, "Integration of Sensors for Fire Fighting Robots," *Journal of Mechatronics and Robotics,* vol. 7, pp. 132-139, 2020.

[27] M. S. Khan, "Fire Detection Systems: A Review," *IEEE Access,* vol. 7, 2019.

[28] H. C. a. D. Zhang, "Thermal Imaging Based Fire Detection Robot,," *Sensors (MDPI),* vol. 18, pp. 1-12, 2018.

[29] P. Y. a. V. Sharma, "Real-Time Control of Fire Fighting Robot Using IoT," *International Journal of Engineering Applied Sciences and Technology,* vol. 4, pp. 90-94, 2019.

[30] M. R. A. M. a. M. S. Hossain, "Autonomous Robotics in Fire Safety Applications," *International Journal of Advanced Computer Science and Applications,* vol. 10, pp. 210-215, 2020.

Appendix A – System Documentation

Introduction

The **Firefighting Robot Using Arduino: Auto Fire Chaser and Extinguisher** is a robotic system designed to autonomously detect and extinguish fire in a given area. The development of this system is motivated by the growing need for automation in fire prevention and emergency response. It minimizes human risk by operating in hazardous environments and reacting rapidly to fire outbreaks.

This robot is equipped with various sensors, including flame sensors and temperature sensors, which enable it to detect fire and respond accordingly. By integrating Arduino microcontrollers, DC motors, and extinguishing mechanisms such as fans or pumps, the robot can navigate the environment, locate the fire source, and activate the extinguishing unit to put out the flame.

The advancement of embedded systems and open-source hardware platforms such as Arduino has enabled the creation of intelligent robotic solutions that are affordable, customizable, and efficient. Communication between components is handled via serial and digital interfaces, ensuring real-time response and control.

This robotic system showcases the potential of integrating sensors, actuators, and control logic to build a practical, real-world application that enhances safety, especially in environments prone to fire hazards such as homes, factories, warehouses, and offices.

Purpose

The purpose of the firefighting robot is to

- **Reduce human risk** by autonomously entering dangerous environments where fire has broken out.
- **Detect fire early** using flame sensors and react before it spreads or causes significant damage.
- **Extinguish fire** using a mounted fan or water spraying system once fire is detected.
- **Navigate autonomously** or semi-autonomously toward the fire source using motor drivers and directional logic.
- **Serve as a prototype** for scalable firefighting robotics systems that could be enhanced for industrial or public safety use.
- **Educate and inspire** engineering and robotics students about real-world applications of embedded systems, automation, and sensor integration.

Appendix B – Design Documentation

Use case diagram

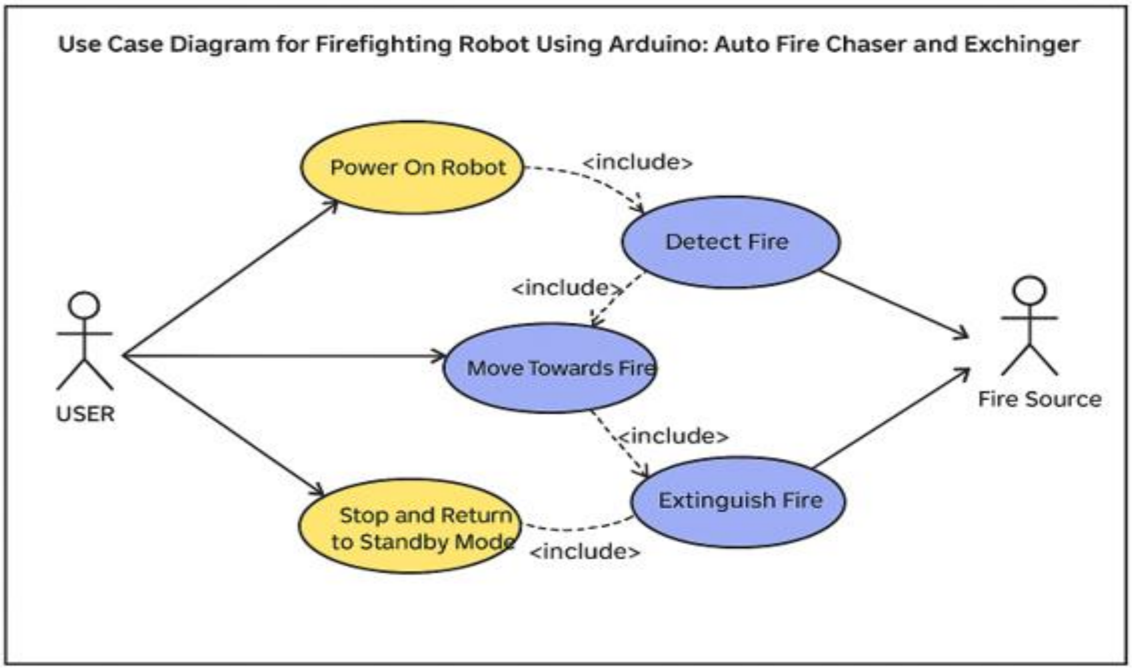


Figure 0.1 Use case Diagram For fire fighting Robot

Class Diagram

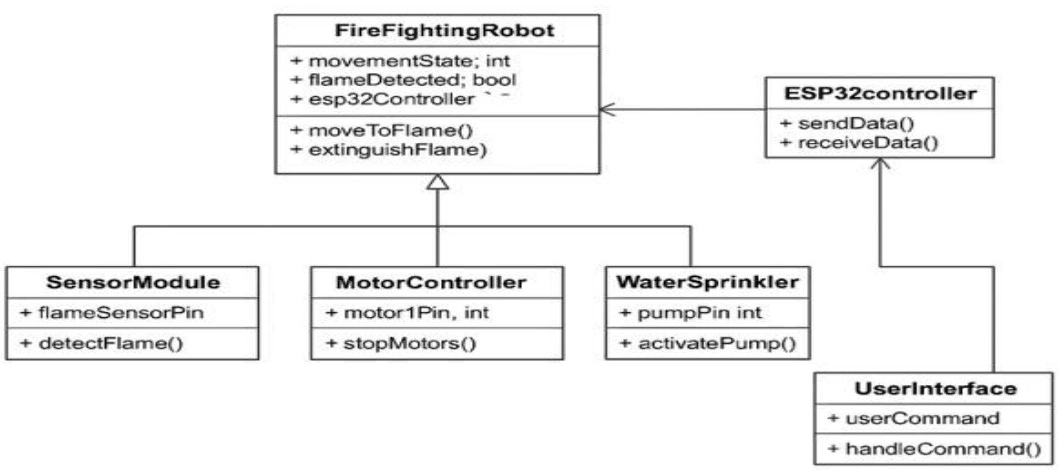


Figure 0.2 Class Diagram for fire fighting robot

Block Diagram

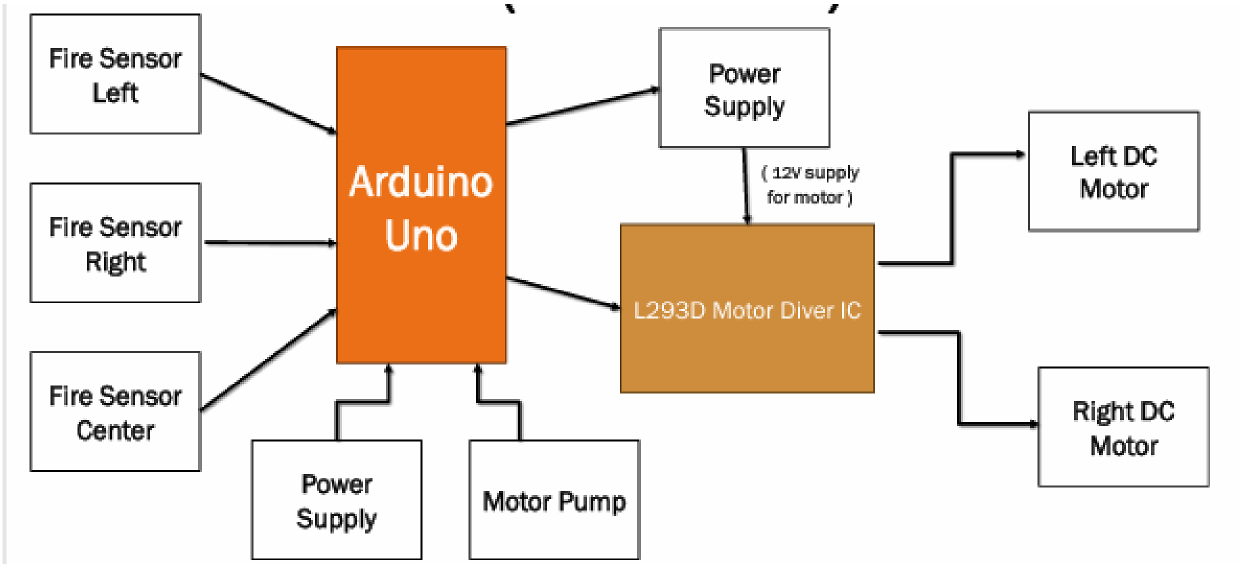


Figure 0.3 Block Diagram for fire fighting Robot

Activity Diagram

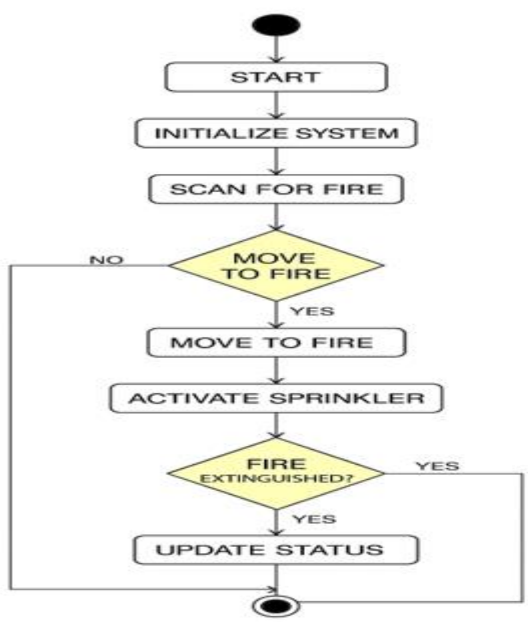


Figure 0.4 Activity Diagram For fire fighting robot

Appendix E – Test Results

The proposed system test result is in the below table

Table 0.1 Test Result

No	Test case ID	Test Case	Result
1	01	Robot detects fire using flame sensor	
2	02	Robot moves towards the detected flame	
3	03	Robot activates water pump to extinguish flame	
4	04	Robot stops when flame is extinguished	
5	05	User presses reset or restart button and robot resume operation	

Appendix F - Code Listing

```
#include <Servo.h>


// Define flame sensor pins
#define FLAME_SENSOR_LEFT A0
#define FLAME_SENSOR_CENTER A1
#define FLAME_SENSOR_RIGHT A2


// Define motor driver pins
#define MOTOR_L1 5  // Left Motor Forward
#define MOTOR_L2 6  // Left Motor Backward
#define MOTOR_R1 9  // Right Motor Forward
#define MOTOR_R2 10 // Right Motor Backward
```

```
// Define relay for water pump
#define RELAY_PIN 7

// Define servo motor pin
#define SERVO_PIN 3

// Create Servo object
Servo waterServo;

// Flame sensor threshold
#define FIRE_DETECTED_THRESHOLD 800  // Adjust based on sensor readings

void setup() {
  // Initialize Serial Monitor
  Serial.begin(9600);

  // Set motor pins as output
  pinMode(MOTOR_L1, OUTPUT);
  pinMode(MOTOR_L2, OUTPUT);
  pinMode(MOTOR_R1, OUTPUT);
  pinMode(MOTOR_R2, OUTPUT);

  // Set relay as output
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);  // Ensure relay (pump) is OFF at startup
```

```
  // Initialize servo motor
  waterServo.attach(SERVO_PIN);
  waterServo.write(90); // Set initial servo position to center

  // Set flame sensor pins as input
  pinMode(FLAME_SENSOR_LEFT, INPUT);
  pinMode(FLAME_SENSOR_CENTER, INPUT);
  pinMode(FLAME_SENSOR_RIGHT, INPUT);

  Serial.println("🚀 Fire Fighting Robot Initialized...");
}
void loop() {
  // Read flame sensor values
  int flameLeft = analogRead(FLAME_SENSOR_LEFT);
  int flameCenter = analogRead(FLAME_SENSOR_CENTER);
  int flameRight = analogRead(FLAME_SENSOR_RIGHT);

  // Print sensor values to Serial Monitor
  Serial.print("Flame Left: ");
  Serial.print(flameLeft);
  Serial.print(" | Flame Center: ");
  Serial.print(flameCenter);
  Serial.print(" | Flame Right: ");
  Serial.println(flameRight);
```

```
    // Check flame detection
    if (flameCenter < FIRE_DETECTED_THRESHOLD) {
        Serial.println("🔥 Fire Detected at Center! Moving Forward...");
        moveForward();
        activateWaterPump();
    }
    else if (flameLeft < FIRE_DETECTED_THRESHOLD) {
        Serial.println("🔥 Fire Detected at Left! Turning Left...");
        turnLeft();
        activateWaterPump();
    }
    else if (flameRight < FIRE_DETECTED_THRESHOLD) {
        Serial.println("🔥 Fire Detected at Right! Turning Right...");
        turnRight();
        activateWaterPump();
    }
    else {
        Serial.println("✅ No Fire Detected. Stopping...");
        stopMotors();
        deactivateWaterPump();  // Ensure pump is OFF when no fire is detected
    }
    delay(500); // Delay for stability
}
// Function to move forward
void moveForward() {
    digitalWrite(MOTOR_L1, HIGH);
    digitalWrite(MOTOR_L2, LOW);
```

```
    digitalWrite(MOTOR_R1, HIGH);
    digitalWrite(MOTOR_R2, LOW);
}
// Function to turn left
void turnLeft() {
    digitalWrite(MOTOR_L1, LOW);
    digitalWrite(MOTOR_L2, HIGH);
    digitalWrite(MOTOR_R1, HIGH);
    digitalWrite(MOTOR_R2, LOW);
    delay(500);  // Adjust timing
    stopMotors();
}
// Function to turn right
void turnRight() {
    digitalWrite(MOTOR_L1, HIGH);
    digitalWrite(MOTOR_L2, LOW);
    digitalWrite(MOTOR_R1, LOW);
    digitalWrite(MOTOR_R2, HIGH);
    delay(500);  // Adjust timing
    stopMotors();
}
// Function to stop motors
void stopMotors() {
    digitalWrite(MOTOR_L1, LOW);
    digitalWrite(MOTOR_L2, LOW);
    digitalWrite(MOTOR_R1, LOW);
    digitalWrite(MOTOR_R2, LOW);
```

```
}

// Function to activate water pump and move servo
void activateWaterPump() {
  Serial.println("🚰 Activating Water Pump...");
  digitalWrite(RELAY_PIN, HIGH);  // Turn ON the water pump
  // Move the servo from left to right
  for (int pos = 60; pos <= 120; pos += 5) {
    waterServo.write(pos);
    delay(100);
  }
  // Move the servo from right to left
  for (int pos = 120; pos >= 60; pos -= 5) {
    waterServo.write(pos);
    delay(100);
  }

  delay(2000);  // Pump stays on for 2 seconds
  deactivateWaterPump();  // Turn off the water pump
}

// Function to deactivate water pump
void deactivateWaterPump() {
  Serial.println("✅ Deactivating Water Pump...");
  digitalWrite(RELAY_PIN, LOW);  // Turn OFF the water pump
  waterServo.write(90);  // Reset servo to center position
}
```

Glossary

Firefighting Robot: A robotic vehicle designed to autonomously detect and extinguish fires using onboard sensors, actuators, and control systems. It operates in environments where human intervention is dangerous or delayed.

Arduino UNO: A microcontroller board based on the ATmega328P. It serves as the brain of the robot, processing sensor inputs and controlling actuators such as motors, pumps, and buzzers.

Flame Sensor: A sensor that detects infrared light emitted by flames. It is used to locate the direction of a fire source for navigation and extinguishing.

Buzzer: An audio signaling device that produces a beeping sound when activated. In the firefighting robot, it is used to issue an audible alert when a fire is detected or when the robot is in emergency mode.

L298N Motor Driver: A dual H-Bridge driver used to control the speed and direction of two DC motors. It interfaces between the Arduino and the motors.

BO Motor (DC Motor): Small brushed DC motors used to move the robot. These are mounted on the chassis and controlled through the motor driver.

Chassis: The physical framework of the robot on which all components are mounted, including wheels, motors, sensors, and electronic modules.

Wheels: Circular mechanical components attached to the motors to provide motion to the robotic car.

Water Pump (5–9V): A miniature pump that is activated by the system to spray water and extinguish detected fires.

Mini Servo Motor: A motor that controls angular movement. It is used to aim the nozzle of the water pump in the direction of the fire.

Relay Module (5V, Single Channel): An electronically controlled switch that allows the low-voltage Arduino to control higher voltage devices like the water pump.

TIP-122 Transistor: A high-gain NPN Darlington transistor used to switch or amplify the signal to drive high-power components like the water pump.

104µF Capacitor: A component used to stabilize voltage and reduce noise in the circuit, ensuring smooth operation of the water pump and motor system.

1K Resistor: A resistor that limits current in the circuit, often used with transistors or LEDs.

Jumper Wires: Flexible electrical wires used to make temporary or permanent connections between components on a breadboard or directly to the Arduino.

Breadboard: A tool for prototyping circuits without soldering. It allows you to test and modify your circuit quickly.

Auto Fire Chasing: The autonomous behavior of the robot to locate the direction of the fire and move toward it based on flame sensor data.

Extinguishing Mechanism: The combination of servo, pump, and control logic that activates water spray in the direction of the fire to suppress it.

Obstacle Avoidance: An optional system where additional sensors detect physical obstacles and reroute the robot accordingly to reach the flame safely.

Power Supply (3.7V Batteries): Rechargeable lithium-ion batteries that provide the required voltage and current for the robot's electronics and motors.

Sensors: Devices that detect changes in the environment (such as fire, light, or distance) and send data to the Arduino for decision-making.

Real-Time Response: The ability of the robot to immediately react to flame detection by triggering motion, alarm, and extinguishing actions.

Autonomous Robot: A robot capable of performing tasks without manual control, based on programming and environmental inputs.

Embedded System: A dedicated computer system designed to perform specific tasks as part of a larger device—in this case, fire detection and response.

Prototyping: The process of creating an early model of the robot to test features and functionality before building the final version.

Emergency Alert System: A component of the robot that issues warnings (via buzzer, LED, or messages) when a fire is detected.

Arduino IDE: The official development environment used to write and upload code to the Arduino board.

Serial Monitor: A tool within the Arduino IDE that displays real-time output from the Arduino for debugging and testing.

Notification System: A system that alerts users of an event. In this robot, the buzzer functions as a key component of the local notification system.